NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

MODELING THE PROGRESSIVE FLOODING CHARACTERISTICS OF THE ARLEIGH BURKE CLASS DESTROYER USING SIMSMART AND EXCEL

by

Keith S. Kulow

June 2000

Thesis Advisor: Charles Calvano
Co-Advisor: Fotis Papoulias

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Modeling the Progressive Flooding Characteristics of the Arleigh Burke Class Destroyer Using SIMSMART and Excel	**5. FUNDING NUMBERS**
6. AUTHOR(S) Kulow, Keith S.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10.SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

The goal of this thesis is to contribute further to the development of a design tool for the modeling of dynamic progressive flooding in ships. In an earlier thesis, LT Thomas Anderson, USN, modeled a generic, mathematically-describable hull form; in this thesis the work is extended by applying his methods and generating new ones in order to accurately model an actual ship hullform, (the Arleigh Burke (DDG-51)), in a progressive flooding scenario. A secondary goal is to create an organized process, complete with any necessary programs or software, which can be applied to <u>any</u> hullform in the future in order to create a progressive flooding model. These goals contribute to the ultimate goal of creating a viable design tool that will allow the Naval Architect to evaluate the potential of a prototype vessel to withstand damage in a progressive flooding scenario.

14. SUBJECT TERMS: Progressive Flooding, Ship Modeling, Naval Architecture, SIMSMART, Ship Damage, Damage Stability	**15. NUMBER OF PAGES** 160
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

MODELING THE PROGRESSIVE FLOODING CHARACTERISTICS OF THE ARLEIGH BURKE CLASS DESTROYER USING SIMSMART AND EXCEL

Keith S. Kulow
Ensign, United States Navy
B.S., United States Naval Academy, 1999

Submitted in partial fulfillment of the
Requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2000

Author: ______________________
Keith S. Kulow

Approved by: ______________________
Charles Calvano, Thesis Advisor

Approved by: ______________________
Fotis Papoulias, Co-Advisor

____
Terry R. McNelley, Chairman
Department of Mechanical Engineering

ABSTRACT

The goal of this thesis is to contribute further to the development of a design tool for the modeling of dynamic progressive flooding in ships. In an earlier thesis, LT Thomas Anderson, USN, modeled a generic, mathematically-describable hull form; in this thesis the work is extended by applying his methods and generating new ones in order to accurately model an actual ship hullform, (the Arleigh Burke (DDG-51)), in a progressive flooding scenario. A secondary goal is to create an organized process, complete with any necessary programs or software, which can be applied to any hullform in the future in order to create a progressive flooding model. These goals contribute to the ultimate goal of creating a viable design tool that will allow the Naval Architect to evaluate the potential of a prototype vessel to withstand damage in a progressive flooding scenario.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The author would like to acknowledge the efforts of a group of individuals who made the creation of this body of work possible. First and foremost, the author would like to acknowledge the efforts of Professor Charles Calvano of the Naval Postgraduate School for being given the opportunity, materials and guidance to contribute to the ongoing design synthesis of the dynamic progressive flooding simulator. In addition, the author would like to acknowledge the efforts of Professor Fotis Papoulias of the Naval Postgraduate School for the programming of Data Input and NPSHS, as well as his advice on the proper utilization of VB as a design tool. This body of work also would not have been possible without the efforts of LCDR David Ruley, who created the SIMSMART model to which the flooding calculator was linked. The author would also like to thank Dr. Greg White and Mrs. Nancy Anderson Harris of the United States Naval Academy for contributing the DDG 51 offsets on which this entire project is based, as well as their advice at the commencement of this thesis on both Excel and FastShip.

I. INTRODUCTION

A. MISSION STATEMENT

The goal of this thesis is to contribute further to the development of a design tool for the modeling of dynamic progressive flooding in ships. In an earlier thesis, LT Thomas Anderson, USN, modeled a generic, mathematically-describable hull form; in this thesis the work is extended by applying his methods and generating new ones in order to accurately model an actual ship hullform, (the Arleigh Burke (DDG-51)), in a progressive flooding scenario. A secondary goal is to create an organized process, complete with any necessary programs or software, which can be applied to <u>any</u> hullform in the future in order to create a progressive flooding model. These goals contribute to the ultimate goal of creating a viable design tool that will allow the Naval Architect to evaluate the potential of a prototype vessel to withstand damage in a progressive flooding scenario. The major computer based tools that will be used to create the model are a fluid dynamics modeling program called SIMSMART, and Microsoft Excel, a spreadsheet based data recording program. In terms of the model itself, this thesis will concentrate on the actual physical representation of the vessel, as well as control of the data flow between the two modeling tools. Most of the ship hardware modeling (piping, pumps, etc.) was accomplished by LCDR David Ruley, USN, and will be detailed in his thesis.

B. BACKGROUND

1. Prior Research

The first official US Navy studies regarding ship damage occurred in the aftermath of the First World War. Up to this point, the process of damage control onboard vessels was primarily an uncoordinated struggle of the sailors versus the sea in a

fight to keep the ship afloat. After the war, the Bureau of Ships realized that setting forth an organized process for conducting damage control would result in a significant increase in ship survivability in battle situations. [Ref. 1] Hence, the science of damage control was born. However, initial efforts towards learning how to save a ship from damage concentrated primarily on organizing damage control efforts of the vessel and the sailors after the damage took place, not designing the ship to help minimize the effect of damage.

It was not until after the Second World War that the idea of designing a vessel from the outset to resist the promulgation of flooding damage became widespread. There was a twofold impetus to this realization: first, the threat of the submarine and torpedo did not truly realize itself until the Second World War, when technology and tactics were refined enough that that threat became significant. Secondly, up until World War II, the idea of designing a ship to resist damage essentially meant putting the thickest skin of armor possible on the sides of the hull of the vessel to prevent holing, primarily by gun projectiles. [Ref. 2] However, over the course of the war, naval weapons were refined to the point that it became unfeasible to try to protect a vessel using an armor skin alone.

In 1947 a design study of vessels that had successfully resisted torpedo casualties was done to determine the extent of damage that they had undergone. The result of this study was the Navy's classic damage control design requirement: that warships should be designed to survive a casualty which produced a hole of a size equal to 15% of the warship's waterline length (with a corresponding value of 12.5% for auxiliaries.) [Ref. 2] Over the following years, additional requirements have been added to the Navy's design philosophy to help refine the designers' approach toward damage. The hole size

requirement has changed for small vessels: for vessels under 100 ft, the hole size cannot exceed one compartment, and vessels that are between 100 ft and 300 ft have a two compartment standard. [Ref. 2] The work of Sarchin and Goldberg in 1962 persuaded the Navy to adopt the 3" margin line for reserve buoyancy that had been set in 1929 for passenger vessels by the International Conference on Safety of Life at Sea (SOLAS). [Ref. 1]. A margin line is a line on the side of a ship hull "defining the highest permissible location . . . of any damaged waterplane in the final condition of sinkage, trim and heel." [Ref. 3] In addition to the 3" margin line, at the time the Navy also adopted a set of standards regarding damaged stability that took transverse flooding, list, roll, and wave motions as well as wind moments into account. These requirements were codified into the Design Data Sheet (DDS) 079-1 design requirement. [Ref. 4] The incorporation of these design requirements has resulted in a significant increase in a vessel's ability to resist damage. However, all of these design requirements deal with the issue of intact damage stability, not dynamic flooding scenarios. In addition, these design requirements are exactly that: guidelines for Naval Architects to follow when they are designing vessels, not tools for the Naval Architect to use to help in the design process. Through the late 1980s, no significant advances in design aids had been made in order to help design and test vessels in a damage stability environment.

The first true tool that was heavily utilized by the Navy in the realm of damage control is the Flooding Casualty Control Software (FCCS.) FCCS began development in the late 1980s, and was initially deployed in 1990 aboard *USS Doyle* (FFG-39.) Following successful evaluations on the fast frigate platform, modules of FCCS were designed for most other ship classes such as DDG 51, DDG 993, CG 47, LHA 1, LHD 1,

LPD 4, and LSD 41. [Ref. 5] The primary goal of the program is use as a tool for the ship's Damage Control Assistant (DCA) to make any number of decisions regarding the operating profile of a vessel at sea. For instance, if a DCA wanted to know if it would be safe for him to conduct small boat operations while towing in Sea State 2, he could insert these parameters into FCCS and it would tell him if those actions would be advisable from a naval architecture standpoint. In addition, FCCS also serves as a virtual logbook; every day the DCA enters how much fuel was burned, food and supplies expended, and waste accumulated. Using this information, FCCS keeps a running tally on the weights and centers of the vessel, as well as it's operating draft and overall supplies. The program also has modules that can account for grounding scenarios, well deck operations, and icing predictions, among other things. [Ref. 5]

The FCCS program is written using FORTRAN 77 and was designed to run in conjunction with Ship Hull Characteristics Program (SHCP). FCCS uses the lost buoyancy method for all intact and damaged stability calculations, and complies with DDS 079-1 requirements. [Ref. 5] However, because the program was designed for use as an on board evaluation tool, its use is limited in the design environment. FCCS cannot conduct real time damage scenarios; it can only predict the final attitude of a vessel that has been damaged based on certain input parameters. In addition, a special module that requires about 1000 man-hours per platform must be specifically designed to run on FCCS before any evaluation can take place. [Ref. 5] Obviously, that requirement is not desirable for a program that will be used as a design tool. While FCCS has proven itself as a valuable tool for a DCA to monitor his ship at sea with, it has not successfully been incorporated into the design process.

The importance of designing a vessel to withstand flooding damage grows as the threats faced in the modern day battlespace change from stand off weapons to those more likely to be encountered in the littorals, such as mines and small craft launched torpedoes. However, a significant advance in the design tools used to meet these ever rising design requirements has not been realized. This dynamic has created a need for a fast, computer based simulation tool that will allow the designer to model a vessel in a progressive flooding scenario that has multiple inputs and multiple scenario parameters. This thesis is an advancement of that effort put forth by Professor Calvano of the Total Ship Systems Engineering Department of the Naval Postgraduate School to design this tool for use in designing the Navy of the next century.

2. Anderson Thesis

This thesis furthers the work of LT Thomas Anderson. His work is detailed in his thesis, entitled "Development of a SIMSMART Based, Progressive Flooding Design Tool." [Ref. 6] LT Anderson laid the groundwork for the creation of a generic design tool by being the first to use SIMSMART as a design tool to model the progressive flooding of a ship. The goals of his project were to set up the program and to prove that SIMSMART mated to Excel could be used as a viable design tool. In addition, he sought to more accurately define the damage that a modern ship would see during combat (as opposed to the antiquated damage profiles that date to World War II era weaponry) and to assess the adequacy of existing dewatering equipment. [Ref. 6]

LT Anderson's thesis made significant advances toward laying the groundwork from which to build a viable design tool. However, the physical ship model that LT Anderson used was extremely simple. The reasoning behind this was to streamline that

portion of the thesis so that he could concentrate on the difficulties that arose in the creation, coordination, and reporting of the simulation timeline. LT Anderson chose the Wigley hullform for his simulations due to the hullform's simplicity. [Ref. 6] The geometric form of a typical ship is so complex that it is impossible to define the surface using a mathematical formula; the order of the equation is simply too high. However, the Wigley hull is a theoretic hullform that is defined by a third order polynomial. Use of this hullform allowed LT Anderson to do all of his hydrostatic calculations using theoretical integrations. Overall, LT Anderson was successful in his endeavor to properly model dynamic progressive flooding. [Ref. 6] While his method was adequate for an initial effort, the use of theoretical integrations to derive hydrostatic properties is not viable for real hullforms. Therefore, an extension of his method for deriving real ship hydrostatic properties must be investigated. In addition, LT Anderson's model was ship-specific. If the SIMSMART/EXCEL method is ever to be used as a design tool, it must be generic so that it can be used to model any number of different hullforms. Generalizing the modeling method and adapting LT Anderson's method to properly model actual vessels forms the crux of the work in this thesis.

C. TOOLS

There are four major computer based tools that are used in the creation of the dynamic progressive flooding model. First and foremost is the fluid dynamic modeling software. For this model, the program selected is called SIMSMART. SIMSMART is a graphically based fluid modeling program created by the Applied High Technology (AHT) Corporation. The program works on the fundamental basis of allowing the user to

use a "grab and drop" interface to create fluid models on a given workspace. Model parts can be fabricated by the user (as the holes to the ocean were for this scenario) or taken directly from generic bins that are supplied with the program and can then be modified to suit the needs of the user. Once the model is made, certain parameters for a scenario are specified, and then the scenario is run. During the course of the scenario, it is possible to alter the status of given parts (e.g. whether a pump is dewatering or not) as well as scenario values (e.g. the amount of water flooding a tank.) SIMSMART is used in this thesis for the dynamic simulation of the progressive flooding scenario. Modeling of the piping and fire main systems for the DDG-51 hullform on SIMSMART was done almost exclusively by LCDR Ruley.

However, using SIMSMART for this particular situation does have one major shortcoming. The program was designed with the intention of using the software to model fluid plants such as canning facilities or breweries. Because of this, the program writers never envisioned a scenario in which the holding tanks (which represent ship compartments in the dynamic progressive flooding model) would "sink" as they filled with fluid, causing the hydrostatic head on the fluid supply to increase. In order to accurately model the ship sinking, a Dynamic Data Exchange (DDE) link must be set up so that another program can accurately model the dynamic sinking and trimming of the ship, in order to update the pressure at the flooding source and, hence, the rate at which water is flooding into the ship. Microsoft Excel was chosen as the program to describe the attitude of the ship over the course of the scenario. Excel was chosen because of its widespread use and ease of operation. In addition, it was determined that the

SIMSMART DDE link was designed to be better suited to work with Excel as opposed to other spreadsheet programs such as Lotus 1-2-3 or QuattroPro.

The third tool that was utilized to create the modeling program was Microsoft Visual Basic 6.0. Visual Basic is an application that allows the user to create executable files (programs) that can be run in a Windows or DOS based environment. Visual Basic was used to create the "front end" program that accepted the user inputs for each particular vessel and then performed the hydrostatic calculations. Visual Basic was chosen in favor of other programming shells such as C++ or COBOL for various reasons. First and foremost, Visual Basic was chosen because it is relatively small and simple to learn and operate as compared to other programming methods. In addition, the possibility exists that Visual Basic and SIMSMART may link up directly via a DDE in the future, and it was desired to keep this option available to future project iterations.

The fourth and final computer based tool that was used extensively over the course of the project was General HydroStatics v. 6.52C (GHS), by Creative Systems, Inc. GHS is an existing, off the shelf ship modeling program that can accept different types of user inputs such as tabular entry, file conversion, and graphics tablet entry. GHS then numerically defines the hullform using the trapezoidal rule over very small increments to obtain a "virtual ship" in its memory. The program can then perform certain static tests on the hullform, and determine certain characteristics of the hullform. Because GHS can solve for static flooding scenarios, it was used primarily as a validation tool to ensure the accuracy of the programs written for the dynamic modeling. GHS was chosen over other existing hydrostatics programs such as Ship Hull Characteristics Program (SHCP) and Advanced Surface Ship Evaluation Tool (ASSET) because of it's

versatility, greater ease of use, and wide acceptance in the field of military as well as general naval architecture.

D. METHODOLOGY

Due to the fact that the goal of this thesis is to create a generic design tool for future use, certain precautions must be taken over the course of the creation and modeling process. When a computer based model is created, it is typically done using one of two methods: either a template program is used in which the user enters certain arguments that are ship specific and the program outputs the model, or the user creates a keel up virtual model that contains all of the nuances of the particular ship that he wishes to model. In either case, the question of overall, generic accuracy of the modeling method is not in question- if a template is used, then accuracy is assumed, and if a specific model is created, the accuracy of the final model is the only concern. In the case of the model made for this thesis, a template was constructed simultaneously with a model of the Arleigh Burke hullform. Creation of a program that will be used extensively in the future requires that a conscious, logical algorithm be taken in the model creation process so that the method's accuracy is verifiable. Despite the fact that the basis of this thesis is program design and not experimentation, it was decided to proceed with the programming process using a method similar to one that would be taken when conducting a scientific experiment. The modeling process was broken down into several segments that will be explained in detail below. Once a step in the model/template creation was completed, it was checked for accuracy. In most instances, GHS was the primary means of data validation. However, some critical steps were checked against

program files that were created using Excel spreadsheets as well. Once it was determined that the step was suitably accurate for not only the particular DDG-51 model but also from a theoretical standpoint, then progress on construction of the model was begun. As each step is explained in greater detail in the text below, the relative errors discovered during the validation process will be discussed. This "step-check" method ensured that the modeling process was accurate, easily verifiable and could be relied upon to provide accurate results when modeling other ships in the future.

There are a few assumptions that were made over the course of the project that should be noted. First and foremost, it should be understood that the modeling that takes place in this thesis is accurate and reliable, yet relatively crude as compared to some of the more advanced naval architecture programs such as GHS. It is understood that a naval architect can use GHS or SHCP to get more accurate compartment calculations than the ones presented here. However, the goal of this thesis is to contribute to the development of a viable and useful design tool that will permit the evaluation of design alternatives, where relative precision counts for more than absolute. Ideally, the work performed in this thesis will be improved upon and refined in a continuing sequence of research. Ultimately, the end result will be a very useful modeling program, though some assumptions made will detract from purely analytical accuracy of the method in order to produce a tool useful for the design modeling process.

The most obvious simplification made at this point, but which is eminently resolvable in future work, is that the effect of transverse bulkheads in the DDG-51 hullform was neglected. As a result of the VLS configuration and the compartmentalization of the forward third of the vessel, there are some transverse

bulkheads in the DDG-51 design. However, the effect of unsymmetrical flooding was deemed too time consuming and complicated at this point in the tool development. Before the method outlined in this thesis becomes an actual design tool, the effect of asymmetric flooding will have to be accounted for to model real world flooding accurately.

A few less significant physical assumptions were also made. The first of these was to neglect the effects of the dynamic movement of water within the physical model of the ship, an effect known as sloshing. The effect of sloshing is relatively minor in reality. That, combined with the complexity of the sloshing calculations led to the effects exclusion in this model. Another simple assumption that was made was to set the permeability of the entire vessel equal to 1.0. This is obviously a worst case scenario approach to scenario modeling, and is not actually correct. However, in order to accurately model the effects of permeability on the flooding compartments, each compartment would have to have an identity that Excel could monitor. Accomplishing this task is as yet beyond the scope of this work. In addition to this simplification, the inner bottoms of the compartments were also ignored. Between the keel of the DDG-51 and the third deck is a vertical watertight deck that acts as an inner bottom to the vessel. Beneath that deck is tankage for the vessel, and ballast, both of which have a variable permeability. The process of accounting for the effects of a vertical watertight bulkhead while accurately modeling a space that has a variable permeability was considered to be properly treatable in future refinements or, perhaps, shown to have a negligible effect. That space was modeled as if it was open space that is part of the compartment above it. Modeling the space in that way essentially models it in a worst case scenario as well (i.e.

the bulkhead is ruptured and there is nothing in the tanks,) which builds another inherent factor of safety into the overall ship assessment.

The final major assumption that was made concerns the ship model itself. The DDG-51 hullform does not fit perfectly into a typical table of offsets. Due to the rake of the bow and the existence of the sonar dome on the bow of the vessel, the DDG-51 hullform has both negative stations as well as negative waterlines. Both of those things were ignored in this model, creating a slightly inaccurate bow shape. However, this inaccuracy is relatively small because the permeability of both of those spaces is extremely low due to equipment storage and the sonar array. Figure (1) is a picture of the DDG-51 hullform outputted from GHS:

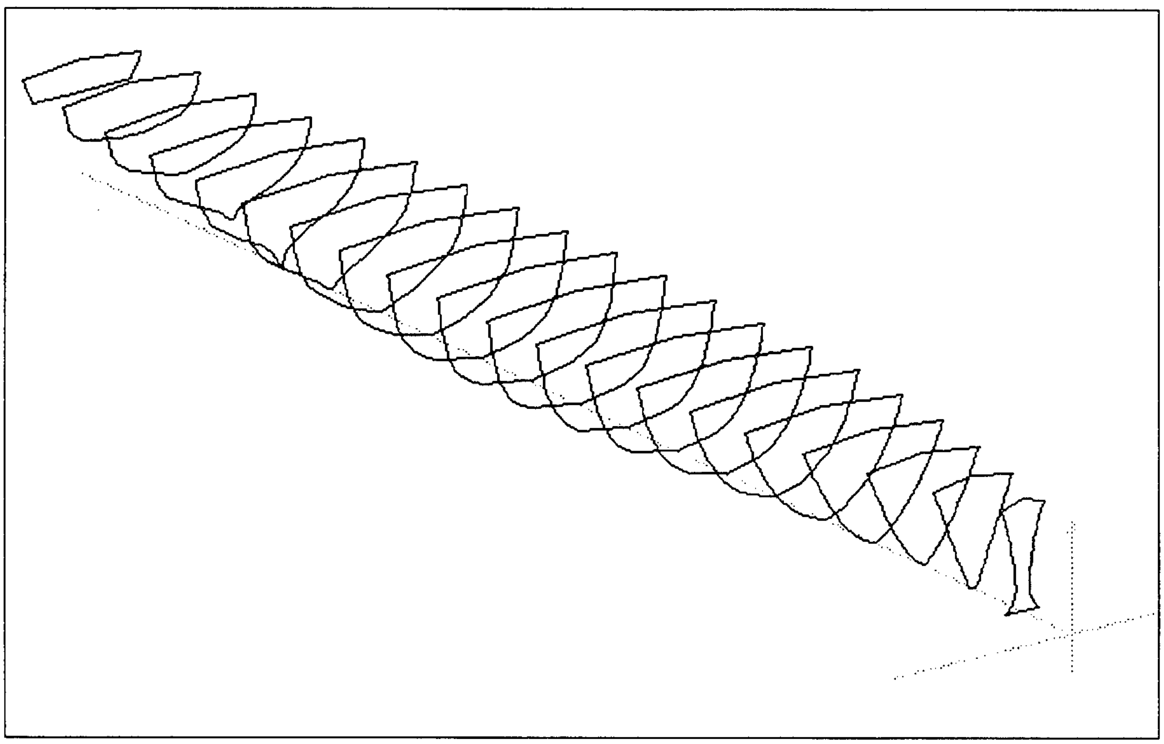

Figure 1: GHS Representation of DDG-51 hullform

II. SHIP DEFINITION

A. ACQUISITION OF OFFSETS

The first step in the process of creating the modeling program was to obtain a viable set of offsets for the DDG-51 hullform. In order to be considered satisfactory, the offset table needed to contain enough detail to accurately create a tabular model of the ship. Yet, at the same time, the table needed to be small enough that it was relatively simple to work with. A table too large would make it difficult to run multiple tests of the program. It was decided that a table with a range of fifteen to forty stations and eleven to twenty nine waterlines would be acceptable for use as a model base.

A model that contained twenty stations and fifteen waterlines was obtained from the Naval Architecture, Ocean and Marine Engineering Department of the US Naval Academy. This model was evaluated by checking a few random offsets against a set of blueprints for DDG-79 that was obtained from Naval Sea Systems Command (NAVSEA). This evaluation determined that the table of offsets was of acceptable accuracy, with a few small noted inaccuracies. The first of these two inaccuracies was that the offset table defined the hull of the vessel up to the 28 foot waterline, whereas the actual hull of the vessel has a deck-at-edge height that ranges from 30 feet at amidships (the lowest point) to 52 feet at the bow. This inaccuracy is relatively insignificant because we are concerned with modeling dynamic progressive flooding, which is a damage scenario in which the height of the floodwater in the ship's compartments is lower than or at equilibrium with the waterline outside the ship. If the water level

reaches 28 feet in any of the compartments, that means that the draft of the ship at that location is 28 feet, therefore the ship is essentially incapacitated.

The second inaccuracy in the table of offsets is that the actual tabular offsets are somewhat inaccurate near the stern. This inaccuracy is relatively insignificant because the volumes of the compartments at the extreme aft end of the vessel are so small that they cannot significantly affect the ship in progressive flooding scenarios. Despite these two inaccuracies, it was determined that the offsets were of usable quality, and would not generate significant errors over the course of the model creation.

B. EXCEL CALCULATION OF PHYSICAL PROPERTIES

Once a set of suitable offsets for DDG-51 was found, the next step was to evaluate these offsets in order to create a physical representation of the ship. Once it was established that creation of a physical representation of the vessel in Excel was actually possible, we could then verify the accuracy using GHS. Next, this model could be used to check the accuracy of the model generated using the front loading program. Due to the mathematical complexity of a real world hullform as opposed to a theoretically shaped hullform, creating an exact representation of the vessel was not a viable option. However, a mathematical model could be constructed with reasonable accuracy using numerical methods in a computer spreadsheet. The idea of using numerical methods to approximate ship hull shapes is not a new process in the field of naval architecture, and there are established formulae and methods governing the creation of a model. These standardized conventions were followed in the course of the creation of the model. Due to the fact that there were an even number of waterplanes, the trapezoidal rule was chosen

as the numerical integration method of choice for the section areas. It is given by the equation:

$$\text{Section Area} = 2 \times \frac{1}{2} \times \text{WS} \times \left(y_1 + 2y_2 + 2y_3 + \ldots + y_n\right) \tag{1}$$

where:
WS = Waterline Spacing
y = Offset

On the other hand, the station spacing was chosen such that there were 21 equally spaced stations along the length of the ship, so the more accurate Simpson's Rule could be utilized for the waterplane integrations. The general form of the Simpson's Rule for numeric integration is:

$$\text{Waterplane Area} = 2 \times \frac{1}{3} \times \text{SS} \times \left(y_1 + 4y_2 + 2y_3 + 4y_4 + 2y_5 + \ldots + 4y_{n-1} + y_n\right) \tag{2}$$

where:
SS = Station Spacing

Numerical integration was used to determine the vessels' longitudinal and vertical volumes, moments about key reference locations such as amidships, and finally, inertial products. Using these values, a set of hydrostatic curves was created. For more information on using general naval architecture principles to generate a set of hydrostatic curves, refer to References [3] or [7].

C. GHS VERIFICATION

Once the initial vessel calculations were completed using Excel, they needed to be verified in order to ensure that using numerical methods on a computerized spreadsheet would be an acceptable means of modeling a vessel shape. GHS was the likely verification choice, due to its simplicity of operation and reliable results. The table of offsets that was used for the Excel calculations was digitized into GHS format. Once

entered into GHS, a regular evaluation of the hullform was performed. This evaluation yielded tables containing the following information:

- Righting Arms versus Heel Angle
- Righting Arms and Areas versus Heel Angle
- Hydrostatic Properties
- Section Areas and Centers
- Various Principal Coefficients

All of this data is organized into Appendix A. With this data, a comprehensive comparison between the results of the GHS output and the Excel results was made. The results of this comparison proved to be very favorable; for example, the vertical center of buoyancy as calculated using Excel was 13.96 feet, whereas it was 13.42 feet according to GHS. Essentially what this means is that Excel and GHS calculated the center of area of a 466 foot long, 28 foot high object to within six inches of each other. The results of comparing the waterplane area calculations between the two methods are shown below in figures (2) and (3):

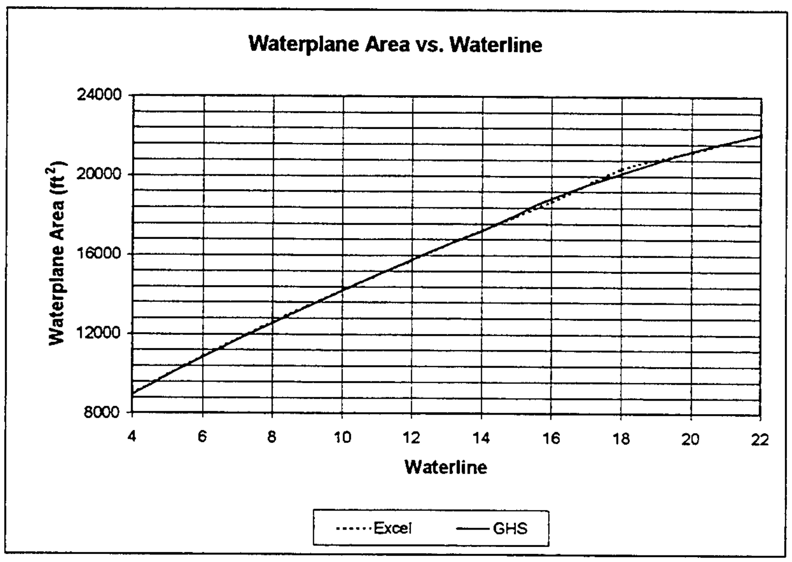

Figure 2: Waterplane Area vs. Waterline

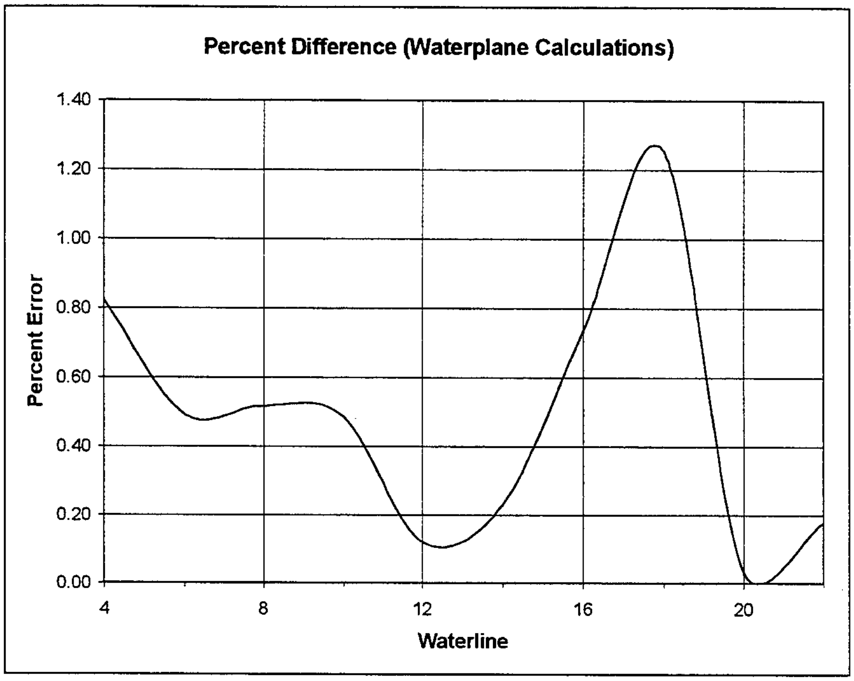

Figure 3: Percent Difference (Waterplane Calculations)

As the two figures show, the waterplane area calculations track together with each other very well, and the percent error between the two is never more than 1.3%. The waterline comparison was only carried out for waterlines 4 through 22 because GHS considers the baseline and the two-foot waterline to be insignificant because of their low areas, so it does not return data for them. Also, it should also be noted that in figure (3), the percent error dips below 0.00% near the twenty-foot waterline. This is an artifact of the automated curve-fairing program in Excel (which was used to create these figures) and could not be avoided. Hand fairing with a French curve would prevent this discrepancy.

The section area calculations between GHS and Excel also compare very well. Figures (4) and (5) show a graphical representation of this comparison.

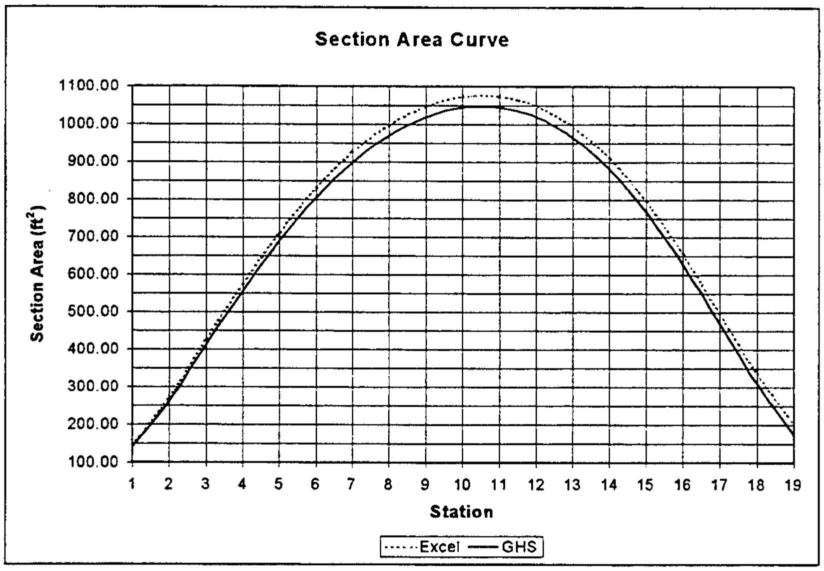

Figure 4: Section Area Curve

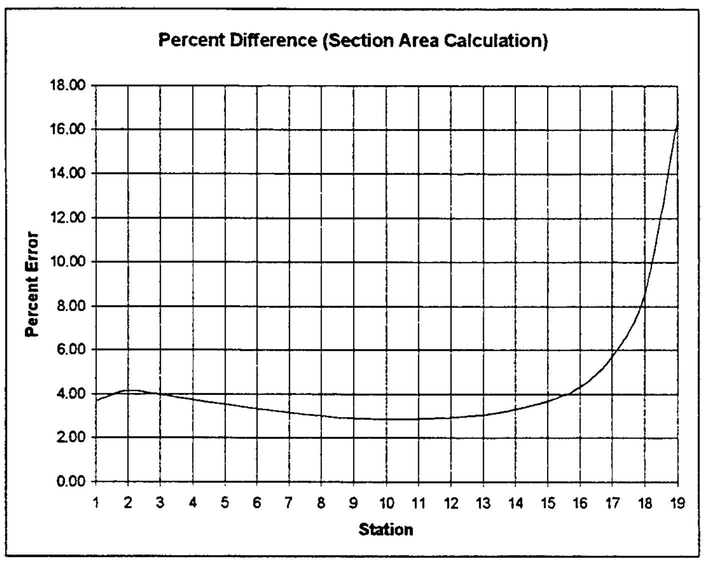

Figure 5: Percent Difference (Section Area Calculation)

Figure (5) shows that the percent difference between the two calculation methods is consistently higher than the error was when the waterplane calculations were done. One explanation for this phenomenon is that all of the station-based calculations were done on Excel using the trapezoidal rule due to the fact that there were an even number of waterlines, whereas the calculations on GHS were performed using Simpson's rule. This

would cause Excel to return slightly higher section areas over the length of the vessel. The graph also shows that the error increases dramatically towards the aft end of the vessel. The larger error at station 19 appears to result from a fairly small error in what is a small number, exaggerating the apparent error expressed as a percentage. It should also be noted that the comparison was not performed at the forward perpendicular or the aft perpendicular where the calculations in GHS become inaccurate.

In summary, the overall comparison turned out very well. The numerical calculations done using Excel were very close to those returned by GHS, which is a validated and respected program. Because of this comparison, it is safe to say that Excel can accurately perform hydrostatic calculations using numerical approximations and is a reliable tool for computerized modeling of a ship hull.

THIS PAGE INTENTIONALLY LEFT BLANK

III. CREATION OF FRONT LOADING PROGRAM

The initial step towards creation of a dynamic progressive flooding simulator was to produce a generic method that could be used to model any vessel electronically. Without this method, the entire vessel model would have to be created from scratch every time a simulation would be run. Creating a mathematical representation of a vessel from scratch is a very drawn out, time consuming process; without successfully automating this process, the idea of creating a generic tool to model progressive flooding was moot. Therefore, the idea of creating a front loading program for the scenario manager was a necessity. It was determined that the starting point for the front loading program would be a table of offsets that defined the vessel that was to be modeled.

A. FIRST ATTEMPT: EXCEL PROGRAM

The initial attempt to create a front loading program for the scenario manager was done using Excel. It seemed to be the most logical tool for creating the front loading program for two reasons: first, because of the numerical nature of the table of offsets, and second because Excel was the program that LT Anderson used to link his scenario manager with SIMSMART. Initially, it was thought that the front loading program and the scenario manager itself would be two different subdivisions of one large Excel workbook. Any sort of dynamic links that would be coordinated between the two subprocceses could be accomplished using Visual Basic for Applications (VBA.) VBA is a simple, graphically based programming suite that accompanies Excel and is designed to incorporate simple automated processes into Excel spreadsheets.

As the design of the front loading progressed, it became apparent that Excel might not be the best tool for the front loading process. The idea was to create the process using DDG-51 as a base, and then "genericize" the process once the program was created. One of the first problems encountered was how to deal with offset tables that could theoretically be of any size. For example, the table of offsets that was used for DDG-51 had 21 stations and 15 waterlines. This table of offsets was of perfectly adequate size to define the DDG-51, but an offset table that size would be a poor choice to model an aircraft carrier, for example, because the model would be extremely coarse and inaccurate. The reverse would be true for a river barge. The obvious solution to this problem was to design for the program to accept a table of offsets of any size, and then let Excel evaluate how many cells it would need to create this table on one of its worksheets. The problem with that solution is that Excel does not have the capability to reason and determine that it needs to create a table of a given size. Excel can use logical operators to determine the values that are placed inside any one cell, but it is not possible to program the tool to utilize a certain number of cells based upon a certain requirement. This capability is also beyond the scope of VBA because VBA is designed to manipulate data that has already been placed in a workbook. Only Visual Basic itself would be capable of generating tables that could be of variable size. Despite this apparently large shortcoming of Excel, the idea of using it for the front loading process was not ruled out, and work was continued on the DDG-51 front loading program.

The next major pitfall that was encountered when using Excel as a front loading program came when the creation of a simple, efficient means for generating the hydrostatic properties of the vessel was attempted. The calculation of the hydrostatic

properties of the DDG-51 hullform had already been performed once, when we evaluated Excel's ability to accurately model a vessel using numerical methods. That process was a very long one; it required that the computer calculate the vessel's properties at every waterline for every station possible. When this task was attempted the second time, during the creation of the front loading program, it was realized that the process would have to be streamlined in order for it to be feasible to use. The act of automating a spreadsheet that uses literally hundreds of thousands of cells all at one time would bring any standard desktop machine to a complete standstill, and the simulation would not operate. In the end, the size of the hydrostatic property calculator was cut down significantly through the use of a few if/then logical operators to determine the proper selection of trapezoidal multipliers for particular area calculations. However, the size of the spreadsheet was still prohibitively large to use as a front loading program. This problem, combined with the issue of being unable to program Excel to handle offset tables of variable size showed that the use of Excel as a means to create a front loading program for the simulation would not be possible, and that other avenues toward the solution should be sought.

B. SECOND ATTEMPT: VISUAL BASIC PROGRAM

The next attempt to create a front loading program was done using Visual Basic because it would be able to handle the issue of dealing with variably sized offset tables. At the same time, VB could be formatted to output files in comma separated value (.csv) format, which is a format that is recognized by Excel. Therefore, VB could theoretically handle user inputs, create a table of offsets, perform hydrostatic calculations, and then

output the results of those calculations to Excel for use in a separate program to deal with changes in the ship's flotation characteristics (trim and drafts) during flooding.

The first front loading program written using Visual Basic was called "Data Input" and was made available by Professor Fotis Papoulias of the Mechanical Engineering Department of the Naval Postgraduate School. The program creates a Windows based executable file to act as a user interface through which a table of offsets could be created, station by station. It starts by querying the user to enter the number of stations and waterlines in the table of offsets. Once this information is entered, the program allows the user to enter the waterline spacing for the vessel. Finally, the program goes along the length of the vessel, asking at each station for a station location with respect to the forward perpendicular (FP), and an array of offsets for each waterline. As this data is being entered, the program is filing the data column by column into a table of offsets that it has constructed based upon the number of stations and waterlines specified by the user.

When all of the offset information has been tabulated, it is saved as a file and the program defaults to a data management window. While there are forms that indicate that many different calculations are available through the data management window, only the hydrostatic property calculator is active. (The other options are reserved for future development.) After a filename is specified, the program will calculate the hydrostatic properties of the vessel in the table of offsets, and output these properties in .csv format to be viewed in Excel. The two outputs from this program, the table of offsets and hydrostatic curves, can then be manually entered into another workbook that can manage the data links to SIMSMART.

In order to ensure computational accuracy, a comparison was drawn between Data Input and GHS. The DDG-51 offsets that were used for the Excel and GHS calculations were entered into Data Input through the user interface, and the hydrostatic properties were calculated. The results were then compared with the hydrostatic properties calculated using GHS. This comparison resulted in figure (6,) which is a representation of the percent difference between the Data Input calculations and the GHS calculations.

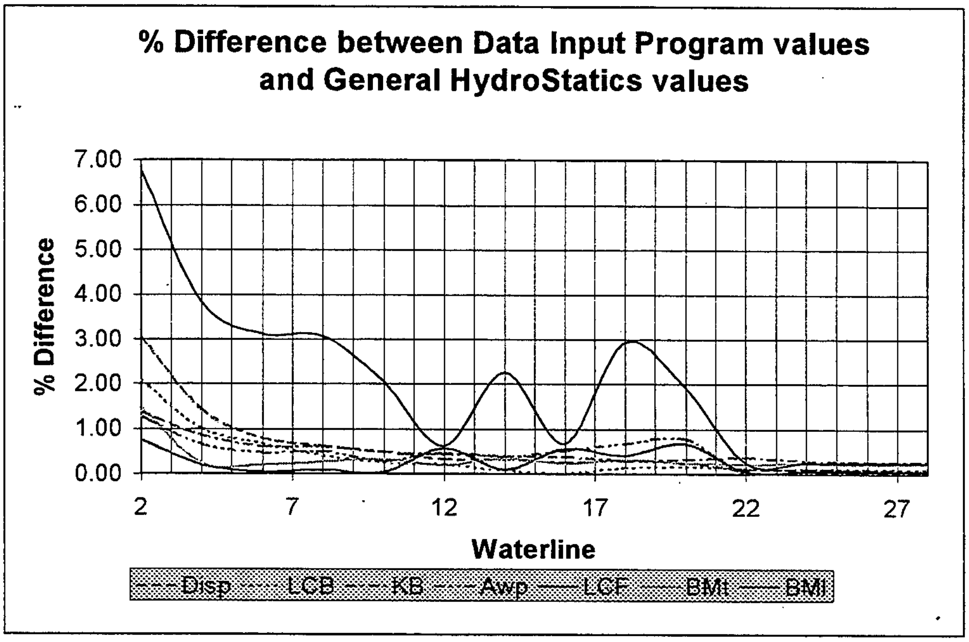

Figure 6: Percent Difference between Data Input values and General HydroStatics values

As figure (6) shows, the calculations performed using Data Input are very close to those created using GHS. Most of the calculations have a percent difference that is less that one percent, which is well within the range of acceptability. However, the longitudinal metacentric height (BM_L) shows a high amount of error as compared to the rest of the values, and it shows some alarming fluctuations from waterline to waterline. Closer inspection also shows that the LCF displays the same fluctuations, but on a smaller scale than is evident with the BM_L. The reason for these fluctuations can be seen when the waterplane area trend is examined. The waterplane area also shows a small

amount of fluctuation, due to the method in which Data Input calculated the waterplane areas at each waterline. Both the LCF and BM_L are based upon integrations of the waterplane areas, and when these integrations are performed, they will have a tendency to magnify any fluctuation in the data. However, even the magnified error of the BM_L is less that three percent in the critical range of waterlines where the ship will most often operate. For this reason, the error is deemed to be acceptable. Overall, the results calculated using the Data Input program can be considered to be accurate enough to be relied upon.

While the Data Input program is most definitely an improvement over the front loading program created using Excel, one major problem was encountered in its utilization. The Data Input program is unable to perform any compartment calculations. The program has the capability to do intermediate hydrostatic calculations; i.e., it has the capacity to determine the hydrostatic properties of a designated piece of the vessel between two designated stations. However, the program can only perform intermediate hydrostatic calculations for stations that are entered as part of the table of offsets. Unfortunately, most vessels are not designed such that their compartment spacing coincides with the hull station spacing. In order to be used as a comprehensive front loading program, the routine must be able to interpolate the offsets of intermediate stations in order to subdivide the vessel into proper compartments. From there, the program must also be able to conduct hydrostatic calculations for these designated compartments. Data Input does not have this capability. Because of this shortcoming, the Data Input program was not a satisfactory solution to meet the needs of a front loading program and had to be modified.

C. THIRD ATTEMPT: NPSHS

The third and final attempt to create a front loading program was also done using Visual Basic. This third program, called Naval Postgraduate School HydroStatics (NPSHS) was created using Data Input as a base. Most of the forms used to create NPSHS were directly taken from Data Input, as was much of the control code. This code can be viewed in Appendix D. Once again, Professor Papoulias was almost solely responsible for the creation of the program. The main difference between NPSHS and Data Input was that the control logic was changed such that NPSHS could interpolate stations between the ones that were entered into the table of offsets. This change allows for NPSHS to create compartments within the overall vessel, and conduct hydrostatic calculations on these compartments. The user interface for entering the table of offsets is identical in the two programs, as is the ship hydrostatic calculator. However, there is a box in the data management window that allows the user to enter the location of longitudinal bulkheads. From inside that window, the user can then conduct calculations on the designated compartments and output these calculations into .csv files.

Because the hydrostatic calculator in NPSHS is the same as the one in Data Input, there is no need to reevaluate it for accuracy. However, in order to maintain consistency, the accuracy of the compartment calculations needs to be checked. This was done by reading three compartments from the plans of the DDG 51 and entering them into NPSHS. The compartments were chosen based on location: one far forward near the bow, one that straddles amidships, and one near the stern. Once these three compartments were entered into NPSHS, the hydrostatic calculations were performed on them. Next, three tanks were built into the existing GHS model of DDG 51. These tanks

corresponded to the three compartments that were modeled using NPSHS. The hydrostatic calculations were then run on the tanks in GHS, and compared with the results of the NPSHS output. Three values for each tank were compared: displacement, LCG, and KG. The results of these comparisons is a series of figures shown below:

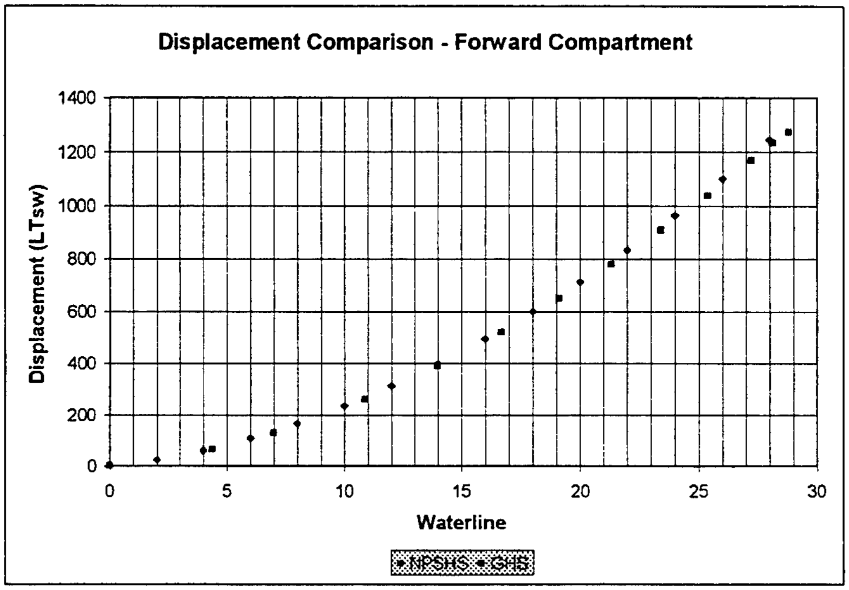

Figure 7: Displacement Comparison- Forward Compartment

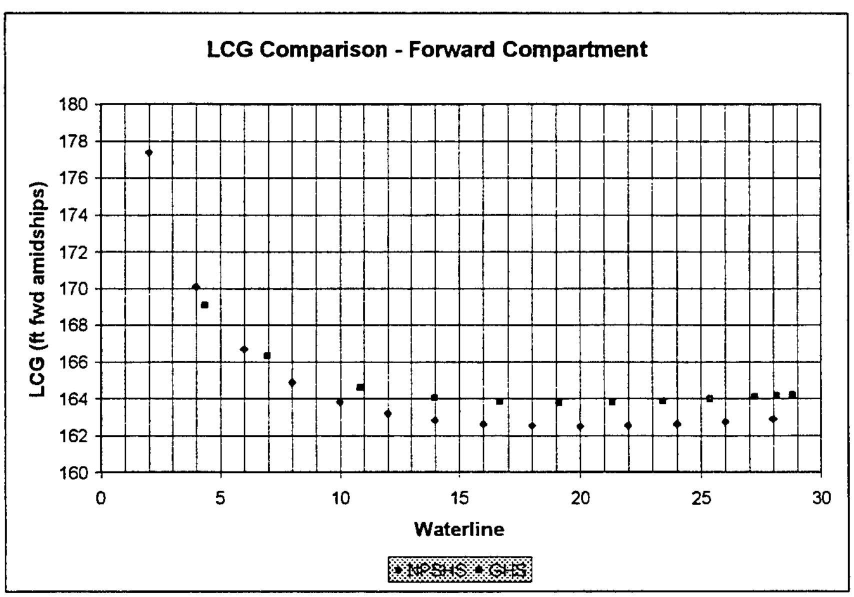

Figure 8: LCG Comparison- Forward Compartment

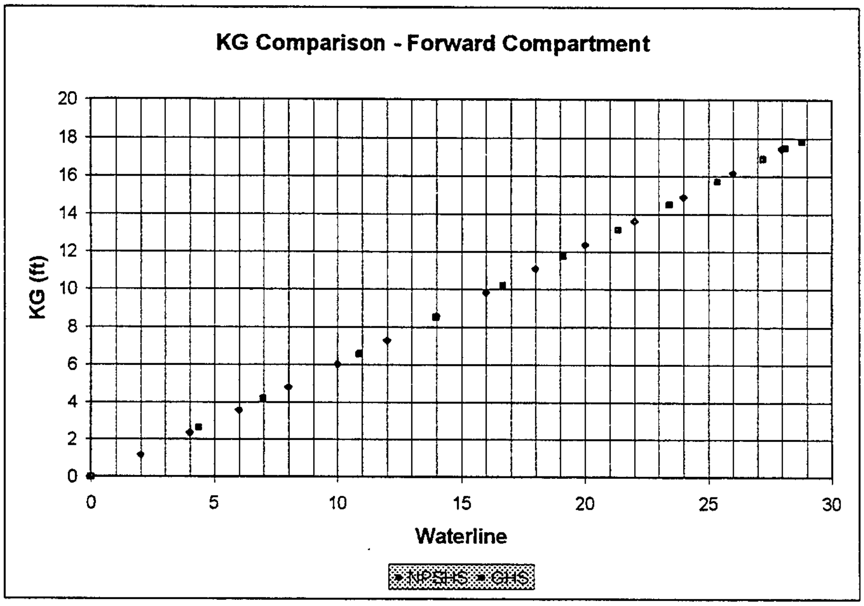

Figure 9: KG Comparison- Amidships Compartment

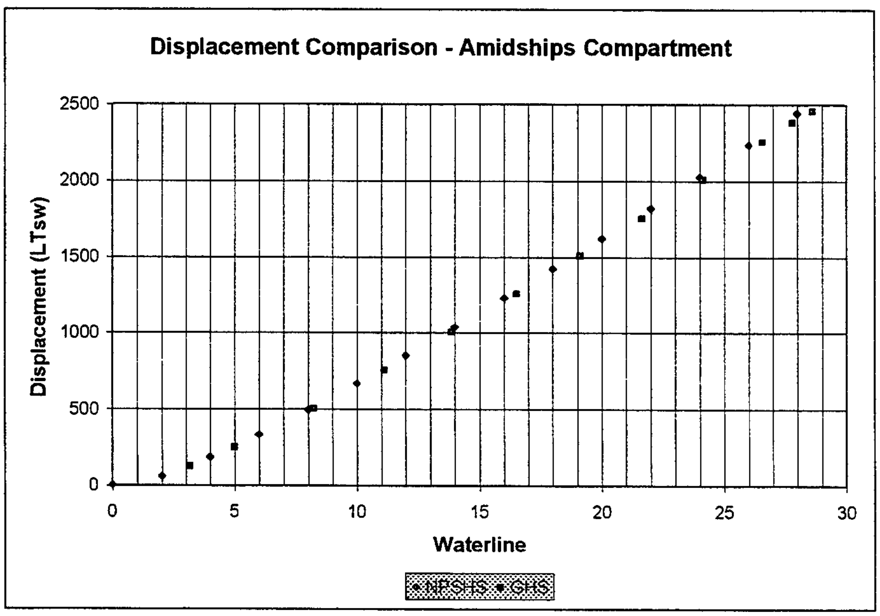

Figure 10: Displacement Comparison

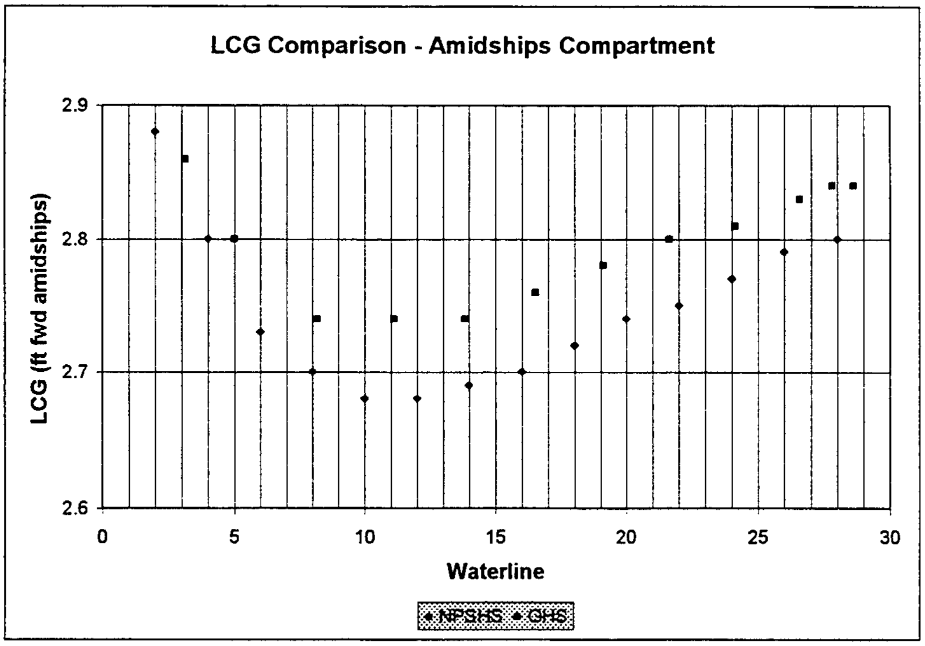

Figure 11: LCG Comparison- Amidships Compartment

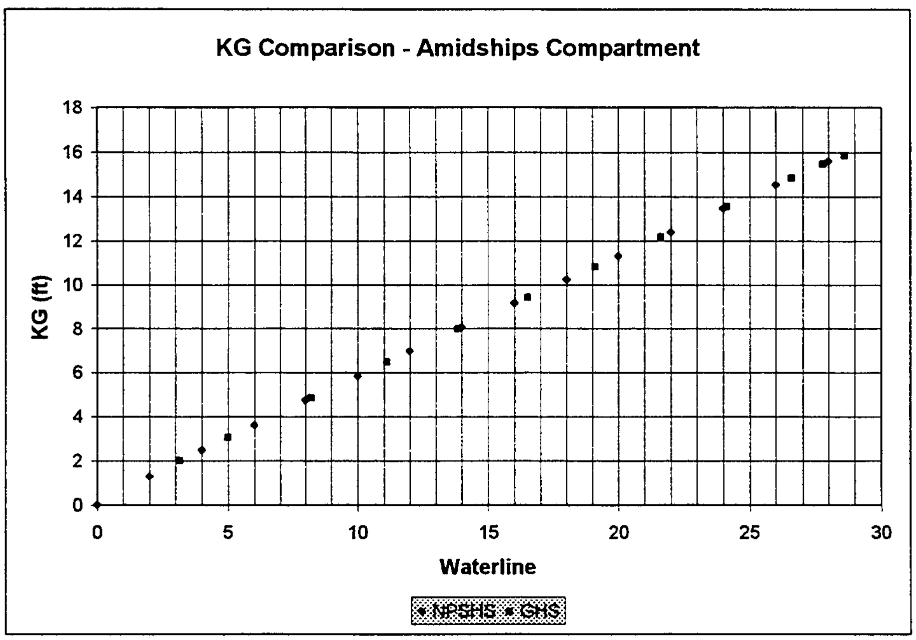

Figure 12: KG Comparison- Amidships Compartment

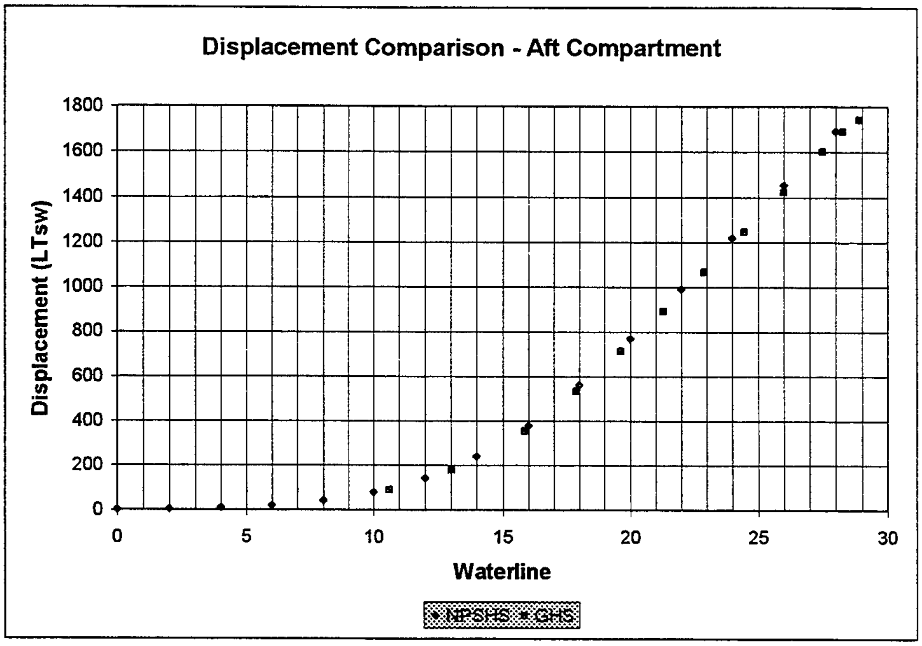

Figure 13: Displacement Comparison- Aft Compartment

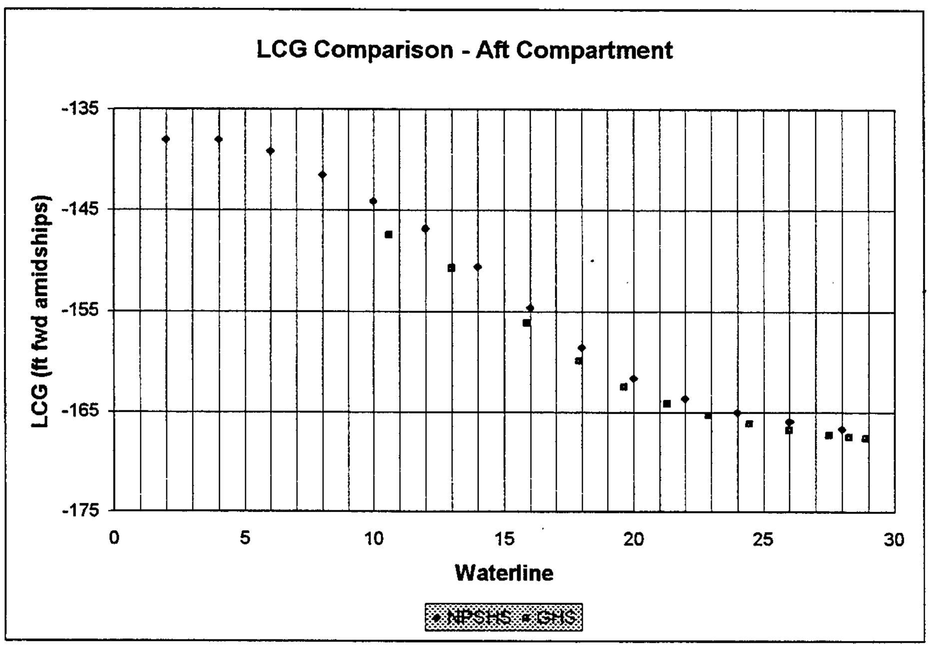

Figure 14: LCG Comparison- Aft Compartment

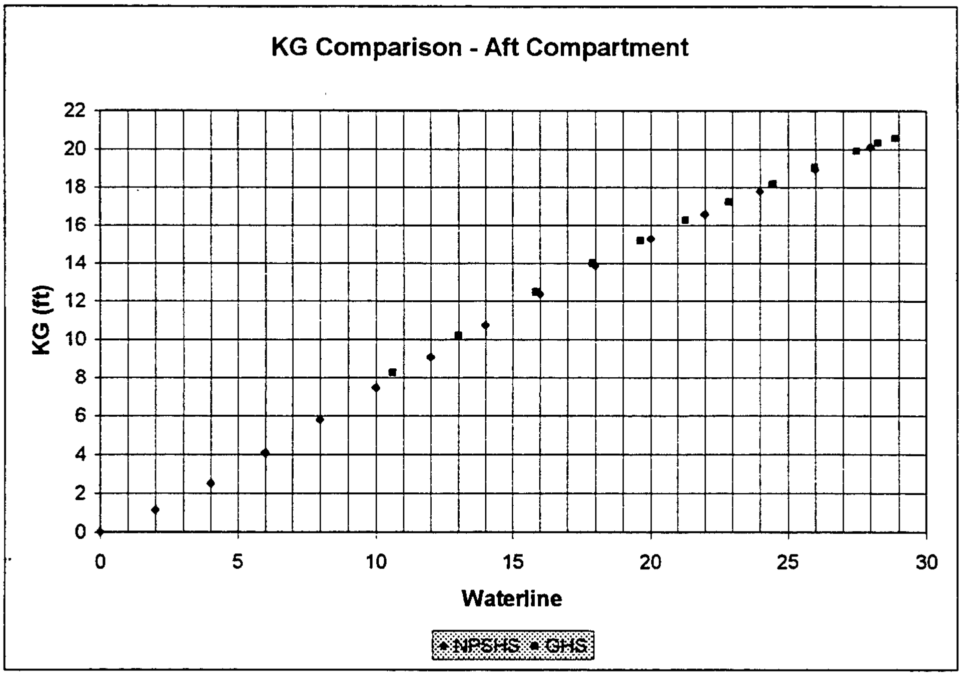

Figure 15: KG Comparison- Aft Compartment

As figures (7) through (15) show, the correlation between GHS and NPSHS for compartment calculations is almost exact. Upon first inspection, it appears that figures (8,) (11,) and (14) show a significant discrepancy in the LCG calculations. However, when the scales of these figures are examined, it becomes apparent that these values are plotted over a much smaller range than the other graphs, and that the accuracy is better than what is initially apparent. It should also be noted that the curves in the comparison figures were not faired, and that percent difference curves were not generated. This is due to the fact that the data used to make the curves were not evenly spaced, and the waterline locations of the two data strings in each graph did not match up to each other. Because of this discrepancy, Excel cannot fair the curves because it does not consider the data to be continuous. The percent difference calculations could not be made because a direct comparison could not be drawn at given waterlines of the data. However, the

overall trends shown in the graphs prove that the two calculation methods correlate well. Thus, NPSHS can be relied upon as an accurate means of conducting compartment calculations and was determined to be an adequate solution to the problem of creating a front loading program for the computer model.

The program proved that it is user friendly, simple in operation, and accurate. It can satisfactorily receive inputs from the user and create a table of offsets; perform hydrostatic calculations on the vessel as a whole as well as on user designated compartments along the length of the vessel; and output the resultant vessel data in a format which is easily digested by Excel. All calculations made by NPSHS were verified using GHS, so its accuracy is reliable. The program meets all of the design criteria set forth in the creation process, so therefore, it is an adequate solution for use as a front loading program. The next step in the modeling process is the creation of the actual Excel based sinking and trimming program.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CREATION OF THE SINKING PROGRAM

A. METHODOLOGY

The creation of the Excel routine that would manage the "sinking" of the vessel as the scenario progressed was perhaps the most crucial portion of the thesis work. There are many reasons as to why this portion of the body of work had to be approached rationally and logically. First and foremost, the constant evaluation and recalculation of the flooding status of the vessel is what makes the scenario dynamic; without this constant cycle of evaluation the model becomes nothing more than an extremely complex damaged stability program. Secondly, the dynamic portion of the modeling process is the most difficult to verify, therefore, it becomes paramount to do it properly and to ensure accuracy. Finally, the Excel routine is also the portion of the project that requires the most independent creative thought because it essentially boils down to the pure design of a control process. For these reasons, it becomes necessary to take a rational approach to the creation of the process. In keeping with this idea, the routine was created in a modular means; i.e. the whole process was broken down into several sub-processes. Each one of these sub-processes was written on a single worksheet of an Excel workbook. The entire routine can be seen in Appendix E. The sub-processes were checked for output accuracy as well as system robustness (resistance to crashing when obviously erroneous data was entered.) Finally, after all the sub-processes were completed and troubleshot, they were assembled together into one complete routine. The last step was to create a "front page" from which all of the crucial data could be monitored and the process could be controlled. The modular design approach serves to ensure the accuracy of the output results of the program. In addition, it provides a system

of organization to the dynamic modeling process. This organization will make the program easier for users to evaluate as well as for future users to understand and improve upon.

B. STARTUP PAGE

The first step in commencing a new simulation is to set up the Excel trimming spreadsheet. The setup process is completed by following the instructions that are delineated on the startup page of the trimming routine. This page comprises Table (1) of Appendix E. Use of this page allows the designer to initialize variables that are ship and scenario specific (such as initial drafts and displacement), and to ensure that all of the automated processes inside of trimming routine are properly set up. Once all of the steps on the startup page have been completed, the user then saves the workbook as it is and exits Excel. From there, the simulation process can begin.

C. SIMSMART INPUT MANAGER

The first step of the actual scenario is to provide the Excel routine with a starting point for its calculation cycle. This was accomplished by importing data from the running SIMSMART simulation into the input manager page of the Excel spreadsheet. The input manager page can be seen in Table (3) of Appendix E. For any generic hullform Excel will need to know four items of data for each compartment that can theoretically be holed in order to properly control the simulation process: the height of the hole above the baseline, the longitudinal location of the hole with respect to amidships, the volume of flooded water in the tank where the hole is located, and the current pressure on the hole. For example, in our model of the DDG-51, there are twelve watertight compartments each with a possible hole, which means that we need to have a

total of forty-eight uplinks to SIMSMART. For each uplink, there is an Excel command that must be entered into the desired formula cell in order to create the actual path for the link. A generic example of the proper command is:

=simlinkser|flowsheetname!'objectname@variablename@variabletype'

For example, in our model the command to obtain the volume of floodwater in Compartment 9 is:

=(simlinkser|PFAustin_1!'Fr338-370@v_tk@s')

These links are controlled by a software package called SMARTAccess that is included in the SIMSMART package supplied by AHT. SMARTAccess includes several forms of input and output flow controllers for managing output from SIMSMART to various different program types. The controller that exchanges data with Excel is called a DDE Server. The DDE Server establishes a continuous network connection between SIMSMART and Excel, allowing for constant update of information. In essence, as the tanks fill in SIMSMART, the volume difference is instantly recorded by Excel. Because the server is a complete package supplied by AHT, its operation is relatively simple: before commencing the simulation or opening Excel, the designer must activate the links by activating the server. Once this process is complete, the server automatically determines what outputs must be controlled and regulates the SIMSMART-to-Excel link. The other primary function that the input manager subprocess performs is the determination of the overall VCG and LCG (vertical and longitudinal location of the center of gravity) as well as I_T (transverse moment of inertia) of the floodwater. Because the input manager has a tabular record of the current floodwater volumes, it made the most sense from an organizational standpoint to determine the centers of gravity and

inertia on that page of the spreadsheet. The centers of gravity for each individual compartment are determined by the curve reader subprocess and will be detailed later in this thesis. The overall VCG and LCG of the flood water were determined using the following relationships:

$$\text{Overall VCG} = \frac{\sum \left(\text{Compartment VCG} \, x \, \text{Compartment Floodwater Volume}\right)}{\sum \text{Compartment Floodwater Volume}} \tag{3}$$

$$\text{Overall LCG} = \frac{\sum \left(\text{Compartment LCG} \, x \, \text{Compartment Floodwater Volume}\right)}{\sum \text{Compartment Floodwater Volume}} \tag{4}$$

$$\text{Overall } I_T = \frac{\sum \left(\text{Compartment } I_T \, x \, \text{Compartment Floodwater Volume}\right)}{\sum \text{Compartment Floodwater Volume}} \tag{5}$$

Once the input manager subprocess tabulates the overall center of gravity for the vessel, it performs a simple system check to provide a small measure of robustness to the system. When the scenario initially commences, flooding has not yet taken place, and therefore, there is no volume of flooded water in any of the tanks. At this point, Excel is then attempting to divide by zero to determine the centers of gravity for the vessel. In order to avoid doing this, the input manager performs a short algorithm of logical operators to see that there is indeed floodwater in the tanks that can be used to perform calculations upon. Starting with the first compartment aft of the forward perpendicular, in the "Data Flow" table of the "Input" sheet, the input manager determines, using if/then logic statements, if there is floodwater in each tank. If there is water in the tank, then it calculates the overall centers of gravity of the vessel. If not, then the manager goes on to the next tank. If the input manager cycles through the entire length of the vessel and determines that there is

no flooding taking place, then it understands that it cannot use the accepted formulae to determine the centers of gravity of the flooding water, but also that the CG of the floodwater is effectively zero. This system check prevents Excel from generating error messages at the commencement of a scenario. Once the input manager calculates the center of gravity of the flooding water in the vessel, it reports that information to the trimming subprocess for use in determining the new waterline of the vessel.

D. HYDROSTATIC CURVE READER

One significant difference between the Wigley hull model employed by LT Anderson and the actual ship data used in this body of work is the nature of the physical properties of the vessel. The Wigley hull was defined by a mathematical formula; therefore its hydrostatic properties could be calculated exactly at any draft using a few reasonably simple equations. However, a real hullform is not so simple, and its hydrostatic properties do not conform to any given trend that would allow for their simple calculation based upon a given draft. Instead, the hydrostatic properties are determined numerically over a set range of drafts on a regular interval, and then recorded in hydrostatic tables, from which the properties can be read at a later time. In order to create a dynamic modeling process, the program must be able to determine the hydrostatic properties of the given vessel automatically over the course of the scenario. This entails directing the computer to "read" data from a table. In addition, while the hydrostatic data for a given hullform is tabulated at a regular interval of drafts, the computer must be able to determine the hydrostatic data at any possible draft that the vessel can float at. The curve reader subprocess controls the reading and interpreting

process that takes place in order to determine the different hydrostatic values for the vessel. It can be seen in Table (5) of Appendix E.

There are three major steps in the curve reading subprocess: determination of the proper draft from which to read the hydrostatic properties, calling up the properties at the correct draft interval which bounds the desired draft, and calculating the properties at the desired draft. In most progressive flooding scenarios, the vessel is trimmed, as opposed to being on an even waterline. When a vessel is trimmed, there are two approaches to determining the hydrostatic properties of the vessel. The first method is to "redraw" a new set of waterlines on the vessel that are angled to meet the desired trim, and recalculate the hydrostatic properties of the vessel from scratch, for what is essentially a different hullform. This is a costly, time consuming process. However, for large angles of trim where the hydrostatic properties are significantly altered due to the trim, this is necessary. The other method of determining the hydrostatic properties of a trimmed vessel is to determine the hydrostatic properties of the vessel as if it was sitting on an even keel at a draft that is midway between the draft at the two perpendiculars. This second method is a much simpler yet less accurate method. However, for small angles of trim such as will be encountered in most progressive flooding scenarios, the simpler method introduces little error. Because of the second method's simplicity and reasonable accuracy, it was chosen as the method the curve reader would use to determine the proper draft for which the hydrostatic properties would be determined.

Once the curve reader has determined the meanline draft that the vessel is floating at, the next step is to determine the desired draft interval from which to do the property interpolation. Excel once again uses if/then logical operators to determine the bounds of

the desired draft. For example, the DDG-51 model has tabulated hydrostatic data for 15 different waterlines on two-foot intervals ranging from the keel (0 ft.) to the 28-foot waterline. In order to determine the two waterlines that bound the desired draft for the interpolation, the curve reader begins with the two-foot waterline and uses an if/then statement to ask itself if the value of the desired draft is greater than or less than two feet. If the draft is greater, the curve reader repeats the question for the next incremental draft, which is the four-foot waterline. The process repeats itself until the curve reader determines that the desired draft is less than the queried draft. When this happens, the curve reader enters the queried draft as the upper bound for the interpolation, and uses the next lower waterline increment as the lower bound for the desired draft. If the curve reader determines that the desired draft falls exactly on one of the draft increments that it has hydrostatic properties for, then it skips the property interpolation process and directly inputs these properties into the flooding calculation. In addition, if the subprocess determines that the desired draft for the interpolation is out of the range of drafts possible for the given hullform (i.e. a negative draft or one over the deck at edge) then the logic operator will return an "Error" message to the main page and the flooding scenario will halt.

Once the curve reader has resolved which tabulated drafts bound the draft for which the hydrostatic properties are desired, it creates a small table entitled "Hydrostatic Values for Upper and Lower Limits." Into this table the curve reader places all of the hydrostatic values for each of the two bounding drafts for the draft interpolation. This process is accomplished using an if/then based logic progression identical to the one that determines the proper draft range. Once this table of hydrostatic data is tabulated, the

curve reader is able to perform a linear interpolation for each value in order to determine the hydrostatic values at the meanline draft. A linear interpolation was chosen over other, higher order approximations due to its simplicity and reasonable accuracy. The formula for the linear interpolation is given below:

$$\mathrm{Val}_i = \left(\mathrm{Val}_h - \mathrm{Val}_l\right)\left(\frac{\mathrm{Draft}_i - \mathrm{Draft}_l}{\mathrm{Draft}_h - \mathrm{Draft}_l}\right) + \mathrm{Val}_l \tag{6}$$

where:
Val_i = Interpolated Value
Val_i = Higher Property Value
Val_i = Lower Property Value
Draft_i = Draft at Interpolation Point
Val_i = Lower Draft
Val_i = Higher Draft
[Ref. 8]

In order to check the accuracy of the linear interpolation, the hydrostatic values for the DDG-51 were plotted and hand faired using a set of French curves. Then, each data point on each of curve was connected using a linear approximation, in order to observe the worst case scenario error between a linear approximation and the exact hydrostatic values for a sample vessel. The results of this comparison show that there was no more than 5% correlation error between the exact hydrostatic values and the linearly approximated values. This is a reasonable error for this application. The results of this interpolation are the hydrostatic values for the vessel in that particular flooding condition, which are then sent to the trimming calculator to determine the new attitude of the vessel.

The other primary function that the curve reader performs is to determine the center of gravity and the transverse moment of inertia of the flooded water for each individual compartment. The premise behind these calculations is much the same as is the hydrostatic property interpolation, except they are done on a volumetric basis instead

of a draft basis. Using the front loading program, the user creates a VCG, LCG, and I_T vs. volume graph for each compartment, and then manually loads those tables into the Hydrostatics page of the workbook. From there, for each compartment, the curve reader uses another series of if/then logic operators to resolve the upper and lower volume based bounds of flood water volume in each tank. Once it has determined these upper and lower bounds, it creates a table that has those values, along with each volume value's corresponding theoretical VCG, LCG and I_T. Because the ship is symmetrical about the centerline, the TCG (transverse center of gravity) is assumed to be zero – that is, on the centerline. Finally, the curve reader performs a linear interpolation in order to determine the VCG, LCG and I_T of flooded water in each individual compartment. Again, the linear interpolation was chosen for it's simplicity and accuracy, and the method's accuracy was checked graphically against a set of hand faired VCG, LCG and I_T vs. Volume curves. The individual compartment centers of gravity and inertia moments are then sent to the input manager so they can be used to determine the overall values of the flooding water.

E. PERPENDICULAR LOCATOR

The next major subprocess that operates within the trimming program is the perpendicular locator subprocess. This subprocess can be seen in Table (8) of Appendix E. As a vessel rises or sinks and trims with displacement change, the longitudinal location of the forward and aft perpendiculars change. In essence, the ship's waterline gets longer or shorter depending on its load profile. When doing trim calculations, this waterline length change becomes significant, and must be determined in order to ensure accuracy of trimming calculations. Due to the complex nature of a real hullform,

determination of the waterline length change must be done using tabular means. Unfortunately, this process is a difficult one to automate, and must be done primarily by the user. The first step is to obtain a profile of the bow and stern of the vessel, and to mark waterlines on the profile over a regular interval. For example, the DDG-51 model was broken down into two-foot increments. Once these increments have been decided upon, measure at each draft the lateral distance the perpendiculars are from the principal perpendiculars, and enter these distances into the table on the "Perpendiculars" page using the proper sign convention. Once this data has been entered into the workbook, the perpendicular locator will be able to resolve the exact location of the perpendiculars based upon the draft of the vessel at that point. This calculation is done once again using a series of if/then logical operators for each draft. The locator will determine which two regular waterlines that the actual draft falls between and call up the upper and lower bounds of the location of the perpendiculars from the manually entered table using more if/then logic progressions. Next, the locator conducts a linear interpolation to determine the location of both the forward and aft perpendicular. The accuracy of the linear interpolation was tested on the DDG-51 hullform using a graphical means similar to that described in the curve reader section above. Results of this accuracy test show that the approximation is a very accurate one over the range that the vessel will primarily operate at. However, it breaks down at extremely low draft levels, where the stern cuts away sharply to allow for the appendages. However, this inaccuracy is insignificant because the vessel will never operate in that draft range under normal conditions, and certainly will not see that operating profile in a progressive flooding scenario. Because of this, it is safe to say that the linear interpolation is satisfactory for use as a perpendicular locator.

Once the perpendicular locator determines the proper location of the perpendiculars, it sends this information to the trimming subprocess.

F. TRIM LINE ADDED WEIGHT CALCULATOR

Quite possibly the most crucial subprocess in the routine is the actual trimming subprocess. This subprocess is the one that brings all of the organized data together and determines the actual attitude of the vessel in the dynamic model. It can be seen in Table (6) of Appendix E. In essence, the trimming subprocess is actually a static damage analysis calculation made dynamic by performing the calculation repeatedly over an infinitesimally small time step. The speed of this time step is not controllable by the user. Rather, the step size is actually regulated by the processing speed of the machine that the simulation is run on: the faster the machine, the smaller the time step. As the simulation progresses, the trimming manager constantly updates the draft at the forward and aft perpendiculars, which allows the hole manager to determine the dynamic head on all of the holes that are currently flooding the vessel.

There are essentially two classic approaches to performing static damage analysis calculations: the trim line added weight method and the lost buoyancy method. The major difference in the two methods is the way in which they treat the flooded water taken on by the vessel in the damage scenario. The lost buoyancy method treats the damage as if the flooded compartments do not and were never a part of the vessel. It assumes that the shape of the waterplane has fundamentally changed, and makes calculations based upon that. In addition, the lost buoyancy method assumes that the water in the flooded compartment has free communication with the sea. The trim line added weight method, on the other hand, assumes that the floodwater takes the form of a

weight that was loaded onto the vessel and does not have communication with the sea. Using the trim line added weight method, the waterplane shape does not change with the addition of damage. In general, the lost buoyancy is regarded as the more accurate method of the two when the extent of flooding is minimal, and the ship is holed near the waterline. In addition, the lost buoyancy method is simpler to use for normal situations. However, for larger extents of flood damage and more complicated flooding scenarios, the assumption that the waterplane shape of the vessel changes breaks down and is invalid, because the changes in the higher order waterplane area based calculations (such as the metacentric heights) become grossly inaccurate. In addition, the lost buoyancy method limits the user to the assumption that flooding holes freely communicate with the sea, which is often times an invalid assumption for damage sustained in the modern battlespace. For these two reasons, the slightly more complicated trim line added weight method is a better choice for the management of the dynamic model, because it produces more accurate results and better manages more complicated flooding scenarios. [Ref. 3]

Most contemporary hydrostatic programs such as GHS or SHCP do not utilize either classic method of damage flooding calculations. Instead, they typically use more complicated calculation routines that have been refined from the basic trim line added weight method. Many of these methods and algorithms are similar to the classic methods, but go about things in a slightly different manner, or take more detail into account in their calculations. For example, because we are not taking into account the effects of longitudinal bulkheads on flooding dynamics, we are not concerned with asymmetric flooding. We have assumed that all flooding is symmetric about the centerline, which is often times not the case. If asymmetric flooding was a concern, then

the combination of trim with list causes the accuracy of the trim line added weight method to break down rather rapidly and this method would not have been a suitable one for our use. However, due to the general simplicity of the flooding scenarios, the trim line added weight method is acceptable as a reasonably accurate means of managing the trim on the vessel.

The method uses simple geometry to determine the new attitude of a ship that has absorbed some form of damage that has caused flooding. The method is simple and its use is widespread; any basic Naval Architecture text such as Principles of Naval Architecture by E. V. Lewis or Basic Ship Theory by E.C. Rawson and K. J. Tupper will explain the method along with the theory behind it in detail. The calculation requires the principal hydrostatics and particulars of the vessel, volume and center of gravity of the flooding water, and current attitude of the vessel. The first step is to determine the amount the vessel would sink if it were on an even keel and the weight was added at the vessel's center of gravity. This value is called the parallel sinkage, and is given (using English units) by:

$$PS = \frac{WA}{TPI \times 12 i n / f} \tag{7}$$

where:
PS = Parallel Sinkage
WA = Weight Added
TPI = Long tons per inch immersion

Once the parallel sinkage has been determined, the next step is to determine the moment that the floodwater is going place about the center of floatation of the vessel. This moment is due to the fact that the weight of the floodwater was not added exactly on the

center of flotation of the vessel. It is called the trimming moment, and is given in the following equation:

$$TM = WA \times (lcg - LCF) \tag{8}$$

where:
TM = Trimming Moment
lcg = Longitudinal Center of Gravity of the flooding water
LCF = Longitudinal Center of Flotation of the vessel

Now that we have determined the magnitude of the moment that will act upon the vessel, we can determine the amount that the vessel will actually trim in response to the moment. This trim is called the trim between principal perpendiculars, and is defined below:

$$TBP = \frac{TM}{MCT1" \times 12 in/ft} \tag{9}$$

where:
TBP = Trim Between Principal Perpendiculars
MCT1" = Moment to Change Trim One Inch

Now that we know the trim between the principal perpendiculars, we can evaluate what the draft change at each of the actual perpendiculars is as a result of the trimming calculations. These two values are given by the following equations:

$$DCT_{fwd} = \left(\frac{\frac{LBP}{2} + LFP - LCF}{LBP} \right) \times TBP \tag{10}$$

$$DCT_{aft} = \left(\frac{\frac{LBP}{2} + LAP + LCF}{LBP} \right) \times TBP \tag{11}$$

where:
DCT_{fwd} = Draft Change at the Forward Perpendicular due to Trim
DCT_{aft} = Draft Change at the Aft Perpendicular due to Trim
LBP = Length Between Principal Perpendiculars
LFP = Location of the Forward Perpendicular
LAP = Location of the Aft Perpendicular

Once these values have been determined, they can be combined along with the parallel sinkage to determine the new draft at the forward and aft perpendiculars, which is the end result of the round of calculations. The respective formulae for these operations are as follows:

$$\text{Draft}_{\text{fwd}} = \text{Original Draft} + \text{PS} + \text{DCT}_{\text{fwd}} \tag{12}$$
$$\text{Draft}_{aft} = \text{Original Draft} + \text{PS} + \text{DCT}_{aft} \tag{13}$$

In essence, the trimming calculator has determined the new waterline that the vessel is floating at, and adjusted its model of the vessel to suit that new waterline. The manager then sends the new draft information on to the output driver so that it can determine the hydrostatic head on the damage point.

G. SIMSMART OUTPUT DRIVER

The next subprocess in the simulation manager is the first step that occurs after the vessel has been trimmed to a new waterline. Now, the routine is preparing the data to be returned to SIMSMART in order to update its simulation with the new vessel properties. The output driver, which can be seen in Table (7) of Appendix E, performs four major functions. First, the driver determines what the new water depth would be over the theoretical hole that was created in each compartment (there are twelve theoretical holes, one in each compartment, in the DDG-51 model.) The output driver accomplishes this using a very simple physical premise. The driver now knows what the draft is at the forward and aft perpendiculars, and it knows the location of each theoretical hole with respect to those perpendiculars. It is now a trivial calculation of the lever rule to determine the draft at each hole using basic geometry. This calculation is given below:

$$D_{hole} = \left[\left(D_{fp} - D_{ap}\right) \times \frac{\left(\frac{LBP}{2} + L_{hole} + L_{ap}\right)}{LBP + L_{fp} + L_{ap}} \right] + D_{ap} \tag{14}$$

where:
D_{hole} = Draft at the hole
D_{fp} = Draft at the forward perpendicular
D_{ap} = Draft at the aft perpendicular
L_{hole} = Longitudinal location of the hole with respect to amidships (positive forward)
L_{fp} = Longitudinal location of the forward perpendicular with respect to amidships (positive forward)
L_{ap} = Longitudinal location of the aft perpendicular with respect to amidships (positive forward)
[Ref. 7]

The next step, now that the output manager has determined what the draft would be at any hole is to determine what the hydrostatic head would be on the hole(s). The output driver performs this routine using a simple two step process. First, it determines what the hole depth would be by subtracting each hole's height above the keel from the hole's draft. Next, now that the driver knows the depth at the hole, it performs a simple pressure calculation, given by:

$$P = \rho g h + p_{atm} \tag{15}$$

where:
P = hydrostatic pressure
ρ = water density
g = gravity
h = hole depth
p_{atm} = atmospheric pressure
[Ref. 8]

to determine what the hydrostatic head on the hole would be, if the hole actually existed.

The third step in the output driver process occurs when output driver knows what the theoretical pressures are on all of the holes in the simulation model. It now needs to determine which of those theoretical holes are actually being used in the simulation that

is being run. The output driver accomplishes that task using a simple comparative logic operator. First, the output driver queries the input manager for the current pressures on each hole. Once the driver knows this information, it asks itself whether or not the current pressure on the hole is higher than ambient pressure. If the pressure is higher, then the driver resolves that the hole must be active. On the other hand, if the current pressure on the hole is lower than ambient, then the hole must be inactive. The output driver then displays the proper output, which is ready to be sent to SIMSMART, in the cells of column H of the "Hole Characteristics Compiler" table. For the holes that the driver determined to be active, this output is the newly calculated hydrostatic head on the hole, and for the inactive holes, the output is 0.00.

The process of managing the attitude of the vessel is all but completed. The final step in the process is to send the array of new pressures on all of the holes, both those that are active and those that are not, back to SIMSMART. This operation is controlled by another DDE link between SIMSMART and Excel. This particular type of DDE link is called a DDE poke command. Due to the fact that the information in this link is controlled by Excel and not SIMSMART, the command structure must be written in Excel and not controlled by a DDE Sever as was the link from SIMSMART into the input manager. In order to accomplish this, a routine was written in Excel using VBA. This routine, which is shown below, is assigned to a button entitled "Initiate DDE Loop" that appears on the "Main" page of the process.

```
Sub poke()
'
' poke macro
' Macro recorded 4/26/00 by LCDR David Ruley and ENS Keith Kulow


For intCtr = 1 To 10
```

```
'Step 5: Establish a conversation between Excel and SIMSMART
SIMSMARTchan = Application.DDEInitiate("simlinkser","PFAustin_1")

PauseTime = 5
Start = Timer

Do While Timer < Start + PauseTime

Loop

'Step 6: Acquire data from Excel
Set H12 = Worksheets("Holes").Range("H12")
Set D37 = Worksheets("Main").Range("D37")
Set D38 = Worksheets("Main").Range("D38")
Set C9 = Worksheets("KGChange").Range("C9")

'Step 7: Send data to SIMSMART
Application.DDEPoke SIMSMARTchan, "Ocean_11-i@p_s@s", H12
Application.DDEPoke SIMSMARTchan, "Fr370-410@manhour@s", D37
Application.DDEPoke SIMSMARTchan, "Fr370-410@eq_mass@s", D38

'Step 8: Terminate the conversation
Application.DDETerminate SIMSMARTchan

'Step 9: Update the Vessel Drafts
Worksheets("Main").Range("E7") = D37
Worksheets("Main").Range("E8") = D38
Worksheets("Main").Range("E3") = C9

'Step 10: Manage Iteration Counter
Set D59 = Worksheets("Main").Range("D59")
Worksheets("Main").Range("D59") = D59 + 1

Next

End Sub
```

When the button is pushed, a control loop is run which reads all of the pressures from Excel, and sends them to the proper locations in the SIMSMART simulation. When this occurs, SIMSMART then increases or decreases the pressure on the hole as needed. The control loop also sends the drafts at the forward and aft perpendiculars into dummy variables in SIMSMART. This allows SIMSMART to collect all of the cumulative scenario data for later compiling and organization. The VBA routine then counts for five seconds, and then updates SIMSMART once again. This process is repeated 10 times

before the routine automatically comes to a halt. A counter at the bottom of the "Main" page shows many total updates have been performed. The user must zero out the cumulative update counter at the commencement of every new SIMSMART simulation. This is accomplished by entering a zero into the cumulative counter cell, which is cell D59 on the "Main" page. Doing this will not harm the program code or alter the counter in any way. This command controls the flow of information from the trimming process back into SIMSMART and makes up the fourth and final function of the output manager subprocess.

H. SIMULATION CONCLUSION

The simulation conclusion is the final aspect that the Excel trimming routine controls. To accurately model a flooding scenario, there exists a need to properly exhibit the possible outcomes. There are four feasible outcomes in a real flooding scenario,. The first possibility is that the flooding is halted, the ship settles to a new waterline, and the crew begins to repair the vessel. Another possibility is that the vessel takes on enough water that the keel cannot support the loading profile that the floodwater creates, and the keel fails. The keel is the backbone of the ship, so when it cracks, the vessel will founder. Another possible outcome that is more likely in a progressive flooding scenario is that the vessel takes on enough water that the margin line submerges. The margin line represents the point from which flooding damage becomes terminal; once it submerges, the ship is lost. [Ref. 3] The final possible outcome to a flooding scenario is that the ship takes on enough water that the transverse metacentric height of the ship is reduced to zero and becomes negative. When this happens, the ship becomes unstable, and any small amount of excitement will cause the vessel to capsize and sink. [Ref. 3] Because

SIMSMART essentially does not "know" that it is modeling a vessel, the scenario outcomes must be modeled using Excel.

The first two modes of scenario conclusion, ship survival and keel failure, can be handled without using Excel. If the designer decides to end the scenario before losing the vessel, i.e. the flooding holes are closed, then they would plug the holes inside the SIMSMART shell. As for the issue of keel rupture, there already exist several different programs that deal with dynamic loading and unloading of vessels. Because a keel failure is not a direct result of progressive flooding, and because adequate software already exists to model structural failure, this mode of ship loss is not considered in this model. That leaves margin line immersion and loss of stability to be modeled in the trimming program.

On the DDG-51 model used in this body of work, the bulkhead deck effectively exists at the 28-foot waterline, because the table of offsets used did not include the offsets for the waterlines that are located above the 28-foot mark. Due to this, the margin line of the vessel was a straight, continuous line that runs the length of the ship at the 27 foot, 9-inch waterline. While this margin line does not correspond with the true margin line of the vessel, it is acceptable for this application. Due to the continuity of the margin line, there are only two places in which the line can initially submerge: at the bow or at the stern. In the case of perfectly parallel sinkage, both points would submerge at the same time. Thus, the issue of margin line immersion was handled rather simply using a logical operator on the "Main" page of the trimming program. Every time the draft is updated to SIMSMART, this operator individually queries the draft at both the bow and the stern using what the Excel programmers call a nested if/then statement, which is essentially

two if/then operators, one within the other. If the draft at either point is above 27.75 feet, then the operator determines that the margin line has been submerged. It then displays a message in the remarks section of the "Main" page telling the user to stop the simulation because the margin line has been submerged and the vessel has sunk. Due to the nature of the nesting of the statements, if both the bow and the stern submerge at the same time, the system will still operate properly and return the indicator.

The issue of stability loss is more complicated to deal with. Ship stability is dependent upon the height of the metacenter above the center of gravity of the vessel. For stability concerns we are interested in the transverse height of the metacenter (GM_T) of the vessel, but not the longitudinal height (GM_L). This is because the longitudinal inertia moment of the ship about a transverse axis is so high that the ship would never "roll" over its bow or stern. However, the transverse inertia moment of the vessel about a longitudinal axis is much lower, so rolling about that axis is an issue. [Ref. 7] The GM_T of a ship is given by the following formula:

$$GM_T = KB + BM_T - KG - \left(\frac{i_T}{V_S}\right) \tag{16}$$

where:
KB = Vertical Center of Buoyancy
BM_T = Distance from the center of buoyancy to the transverse metacenter
KG = Center of gravity of the ship
i_T = Transverse inertia moment of floodwater
V_S = Underwater volume of ship
[Ref. 3]

As the vessel sinks, the KB of the vessel rises, and the BM_T lowers. The KG of the vessel changes according to the following equation:

$$KG_{CHANGED} = \frac{(\Delta \times KG) + \sum (kg \times w)}{\Delta + w} \tag{17}$$

where:
Δ = Ship Displacement
kg = vertical center of gravity of compartment floodwater
w = weight of compartment floodwater
[Ref. 7]

Also, as the ship floods, the i_T of the vessel rises with respect to the underwater volume of the ship. This effect is called the free surface effect, and occurs as a result of the movement of the water inside the vessel. The movement of the water has a tendency to magnify any heeling moments that act upon the ship. [Ref. 3] The combined result of these effects is that as the ship floods, GM_T of the vessel lowers. When GM_T reaches zero and the metacenter dips below the center of gravity of the vessel, the ship becomes unstable. In other words, if any heeling moment is placed on the ship, the ship will have negative stability, and it will capsize.

The page entitled "KGChange," which can be seen in Table (10) of Appendix E, shows the code which monitors the current KG of the ship. This page collects the data necessary to determine the new KG of the vessel after each time step. Once the new KG has been determined, the routine sends the new value to the "Hydrostatics" page of the workbook in order to determine the new hydrostatic properties of the vessel. It also sends the new KG to the free surface calculator. The free surface calculator, which is located on the "FreeSurface" page of the trimming routine, works in much the same manner. It can be seen in Table (9) of Appendix E. The free surface calculator collects the data necessary to determine the current GM_T of the vessel. If, at any point in the simulation, GM_T of the vessel goes to zero, the free surface calculator sends a message to the remarks

section of the "Main" page informing the user that the ship has become unstable and that the simulation should be terminated.

In summary, the simulation can end in three different ways. If the user determines that the ship has survived the flooding casualty, then they can end the scenario by closing the flood holes in SIMSMART. If the margin line of the vessel submerges or the ship becomes unstable, Excel will display a message to the user in the remarks section indicating that the ship has succumbed to flooding and that the simulation should be terminated.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

A. ACCOMPLISHMENTS

The goal of this thesis is to contribute further to the development of a design tool for the modeling of dynamic progressive flooding in ships by applying LT Anderson's methods and generating new ones in order to accurately model the Arleigh Burke (DDG-51) hullform in a progressive flooding scenario. The secondary goal was to create an organized process, complete with any necessary programs or software, which can be applied to any generic hullform in the future in order to create a progressive flooding model. Both of these goals were realized to a reasonable degree over the course of the thesis work. The Visual Basic program NPSHS is a reliable and adequate means of easily converting a vessel shape into a computerized model. The trimming process Excel workbook is a simple, well organized tool which allows the user to manage input from SIMSMART, trim and list the digitized vessel model, and output new scenario parameters back into SIMSMART. The overall process was validated using GHS and was found to be very accurate in its calculation methods. Finally, the process was created such that it is generic, and can be easily modified to accommodate any hullform. Therefore, the tool is viable for use in a design environment.

B. NEXT STEP

Before the tool created in this body of work can be truly utilized, some refinement needs to take place. First and foremost, this design must be applied to a comprehensive set of progressive flooding scenarios. These tests will be conducted by LCDR Ruley, and

will be explained in his thesis. Pending a positive outcome of these tests, there are other refinements to the program that should take place.

The most critical shortcoming of the program that needs to be addressed is the incorporation of some significant stability considerations that were neglected in this initial modeling attempt. In order for this tool to be viable in real world design, it absolutely must take heel and roll into account. To this end, the program should be altered to incorporate longitudinal bulkheads in the model creation and evaluation process.. Of course, if longitudinal bulkheads are incorporated in design, then the program must be refined to include asymmetric flooding in the model.

There are also lesser hydrostatic considerations that should eventually be taken into account. For instance, the sloshing effects of the floodwater can be significant in the dynamic flooding environment and are worthy of investigation. In addition, permeability effects in the compartment calculations should not be ignored. Proper consideration of permeability may well entail the utilization of a variable permeability parameter in the simulation control flow in order to properly model tankage and ballast.

There are also significant improvements that can be made on the overall design process that is presented here. For instance, there still exists a bridge between the output from the NPSHS program and the input of the Excel trimming routine. Currently, the user has to "cut and paste" all of the values from NPSHS into Excel, and modify some minor parameters in Excel to properly run the program. A possible improvement would be to create a module that automatically collected the output data from NPSHS and entered the data into the proper locations in the trimming routine. To this end, over the course of the design process, it was discovered that it might be possible to link

SIMSMART directly to a Visual Basic based executable program such as NPSHS that could manage the sinking and trimming of the vessel. Effectively, this would eliminate the need to use Excel as a middleman between SIMSMART and VB. Streamlining and automating the process in this way could prove to be extremely beneficial to the designer. This possible course of action is worth investigation.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: GENERAL HYDROSTATICS OUTPUTS FOR DDG-51

00-04-03 22:33:01 Page 1
GHS 6.52C DDG51A KEITH

RIGHTING ARMS vs HEEL ANGLE

LCG = 3.00f TCG = 0.00 VCG = 10.00

Origin Depth	Degrees of Trim	Heel	Displacement Weight (LT)	Righting Arms in Trim	in Heel
288.507	87.86f	0.00	8,674.91	0.00	0.000
288.549	87.89f	10.00s	8,674.79	0.00	0.043s
288.567	87.99f	20.00s	8,674.67	0.00	0.082s
288.542	88.15f	30.00s	8,674.67	0.00	0.110s
288.501	88.37f	40.00s	8,675.46	0.00	0.125s
288.447	88.49f	44.90s	8,674.83	0.00	0.126s
288.403	88.63f	50.00s	8,674.83	0.00	0.124s
288.310	88.94f	60.00s	8,674.64	0.00	0.109s

Distances in FEET.--Specific Gravity = 1.025.----------

RIGHTING ARMS vs HEEL ANGLE

Fixed CG: LCG = 3.00f TCG = 0.00 VCG = 10.00

Origin Depth	Degrees of Trim	Heel	Displacement Weight (LT)	Righting Arms in Trim	in Heel	--> Area
320.590	88.20f	0.00	9,964.42	0.00	0.000	0.00
320.625	88.23f	10.00s	9,965.52	0.00	0.031s	0.16
320.659	88.31f	20.00s	9,964.17	0.00	0.058s	0.60
320.634	88.45f	30.00s	9,963.82	0.00	0.079s	1.29
320.588	88.63f	40.00s	9,964.31	0.00	0.089s	2.14
320.523	88.73f	44.90s	9,963.25	0.00	0.091s	2.58
320.509	88.85f	50.00s	9,964.32	0.00	0.089s	3.04
320.412	89.11f	60.00s	9,963.80	0.00	0.078s	3.89

Distances in FEET.--Specific Gravity = 1.025.----Area in Ft-Deg.

Note: The Center of Gravity shown above is for the Fixed Weight of 8674.83 LT. As the tank load centers shift with heel and trim, the total Center of Gravity varies. The righting arms shown above include the effect of the C.G. variation.

HYDROSTATIC PROPERTIES
No Trim, No Heel

Origin Depth	Displacement Weight (LT)	Center of Buoyancy LCB	TCB	VCB	WPA	LCF	BML	BMT
4.000	645.52	215.69a	0.00	2.49	8902	218.81a	2443.3	34.95
5.000	914.47	216.95a	0.00	3.08	9856	220.38a	2007.7	31.30
6.000	1,211.30	218.11a	0.00	3.68	10819	221.86a	1734.7	29.59
7.000	1,533.11	219.09a	0.00	4.28	11677	223.79a	1558.9	27.61
8.000	1,879.38	220.08a	0.00	4.88	12530	225.40a	1422.0	26.47
9.000	2,249.36	221.10a	0.00	5.48	13324	227.43a	1326.1	25.15
10.000	2,642.52	222.18a	0.00	6.08	14141	229.54a	1253.2	24.32
11.000	3,058.44	223.32a	0.00	6.68	14927	232.15a	1203.7	23.34
12.000	3,496.85	224.54a	0.00	7.29	15758	235.01a	1169.5	22.73
13.000	3,954.76	225.69a	0.00	7.90	16380	236.34a	1108.0	21.85
14.000	4,435.35	226.97a	0.00	8.51	17212	239.79a	1097.2	21.32
15.000	4,938.55	228.37a	0.00	9.12	17973	243.05a	1085.3	20.61
16.000	5,464.05	229.87a	0.00	9.74	18827	247.03a	1088.3	20.21
17.000	6,006.84	231.29a	0.00	10.35	19297	247.63a	1036.0	19.38
18.000	6,570.03	232.76a	0.00	10.97	20079	251.18a	1038.5	18.97
19.000	7,151.92	234.24a	0.00	11.58	20627	252.82a	1012.4	18.29
20.000	7,749.73	235.64a	0.00	12.20	21206	254.68a	993.5	17.75
21.000	8,362.85	236.99a	0.00	12.81	21703	256.05a	969.0	17.14
22.000	8,989.28	238.28a	0.00	13.42	22072	256.08a	931.5	16.50
23.000	9,623.98	239.44a	0.00	14.02	22366	255.65a	889.6	15.91
24.000	10,267.08	240.44a	0.00	14.61	22660	255.24a	852.1	15.38
25.000	10,918.31	241.31a	0.00	15.21	22935	254.74a	817.4	14.89
26.000	11,577.38	242.06a	0.00	15.79	23209	254.25a	786.1	14.45
27.000	12,244.18	242.71a	0.00	16.38	23476	253.70a	757.5	14.04
28.000	12,918.60	243.27a	0.00	16.96	23742	253.16a	731.4	13.67

Distances in FEET.-------Specific Gravity = 1.025.--------------------------

Part: HULL Component: HULL.C Side: CL Effectiveness: 1.000
Origin Depth: 21.500 Trim: zero Heel: zero

HULL.C COMPONENT SECTIONS

Section Location	Baseline Depth	Section Area	TCtr	VCtr	Waterline Width	Ctr
0.00	21.50					
0.14a	21.50	0.00	0.00	21.50	0.00	0.00
23.30a	21.50	139.83	0.00	11.41	10.00	0.00
46.60a	21.50	260.27	0.00	13.47	20.96	0.00
69.90a	21.50	410.39	0.00	13.51	32.06	0.00
93.20a	21.50	553.93	0.00	13.33	41.40	0.00
116.50a	21.50	686.99	0.00	13.09	48.20	0.00
139.80a	21.50	803.83	0.00	12.79	53.00	0.00
163.10a	21.50	897.24	0.00	12.49	56.08	0.00
186.40a	21.50	967.42	0.00	12.27	57.80	0.00
209.70a	21.50	1017.18	0.00	12.13	58.76	0.00
233.00a	21.50	1043.01	0.00	12.07	59.20	0.00
256.30a	21.50	1043.52	0.00	12.11	59.28	0.00
279.60a	21.50	1018.27	0.00	12.26	59.16	0.00
302.90a	21.50	964.17	0.00	12.56	58.76	0.00
326.20a	21.50	880.09	0.00	13.04	57.90	0.00
349.50a	21.50	766.99	0.00	13.72	56.48	0.00
372.80a	21.50	628.28	0.00	14.72	54.40	0.00
396.10a	21.50	469.18	0.00	16.23	51.40	0.00
419.40a	21.50	309.98	0.00	17.81	47.12	0.00
442.70a	21.50	173.29	0.00	19.22	42.00	0.00
466.00a	21.50	0.00	0.00	21.50	36.88	0.00
466.00a	21.50	0.00	0.00	21.50		

Distances in FEET.

Part: HULL Component: HULL.C Side: CL Effectiveness:
1.000

Origin Depth: 21.500 Trim: zero Heel: zero

HULL.C COMPONENT FORM

Volume = 303680 Cubic Ft LCB = 237.64a TCB = 0.00 VCB = 13.11

B L O C K D I M E N S I O N S
Length = 465.86 Breadth = 59.28 Depth (deepest point) = 21.50
Length/Breadth = 7.86 Length/Depth = 21.67 Breadth/Depth = 2.757
Breadth - Length/10 = 12.69 Ft Block Coefficient = 0.511
Displacement-Length Ratio = 85.8 Length-Volume Ratio = 6.93

W A T E R P L A N E
Area = 21957.3 Square Ft LCA = 256.77 TCA = 0.00
Moments of Inertia: IL = 2.911E+08 Ft^4 IT = 5.125E+06 Ft^4
Length = 465.86 Breadth = 59.28 Waterplane Coefficient = 0.795

M A X I M U M S E C T I O N
Area = 1043.5 Square Ft Coefficient = 0.819

P R I S M A T I C C O E F F I C I E N T S
Cp = 0.625 Cvp = 0.643

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: NPSHS OUTPUTS FOR DDG-51 HULLFORM

PART 1: NPSHS Generated Table of Offsets

15	21													
	0	2	4	6	8	10	12	14	16	18	20	22	24	26
0	0	0	0	0	0	0	0	0	0	0	0	0.16	0.48	0.87
23.3	5.64	3.48	2.43	2.18	2.27	2.49	2.78	3.12	3.53	4.01	4.56	5.16	5.81	6.51
46.6	1.18	2.2	3.21	4.11	4.96	5.81	6.65	7.48	8.31	9.12	9.94	10.77	11.62	12.49
69.9	0.2	3.27	5.1	6.68	8.14	9.53	10.83	12.06	13.2	14.28	15.31	16.31	17.3	18.27
93.2	0.73	4.65	7.22	9.41	11.35	13.12	14.74	16.21	17.53	18.74	19.87	20.93	21.93	22.91
116.5	0.99	6.39	9.69	12.38	14.68	16.69	18.42	19.92	21.21	22.36	23.4	24.37	25.28	26.15
139.8	1	8.79	12.73	15.6	17.89	19.79	21.39	22.75	23.93	24.98	25.91	26.75	27.52	28.24
163.1	1	11.7	15.76	18.43	20.49	22.15	23.54	24.72	25.77	26.68	27.49	28.21	28.84	29.41
186.4	1	14.21	18.11	20.59	22.48	23.98	25.2	26.21	27.06	27.79	28.44	29	29.52	29.98
209.7	1	15.92	19.76	22.15	23.94	25.34	26.44	27.3	27.98	28.54	29.03	29.48	29.91	30.31
233	1	16.55	20.56	23.01	24.78	26.12	27.14	27.9	28.48	28.94	29.34	29.72	30.09	30.46
256.3	1	15.9	20.38	23.05	24.92	26.29	27.31	28.05	28.6	29.03	29.41	29.76	30.11	30.47
279.6	1	13.9	18.99	22.11	24.28	25.87	27.02	27.84	28.44	28.9	29.29	29.66	30.01	30.37
302.9	1	10.87	16.09	19.78	22.54	24.66	26.22	27.3	28.05	28.6	29.05	29.44	29.82	30.19
326.2	1	7.34	12.1	16.11	19.46	22.34	24.65	26.27	27.32	28.05	28.6	29.05	29.47	29.86
349.5	0.98	3.74	7.4	11.5	15.38	18.88	22	24.39	26.01	27.05	27.79	28.36	28.84	29.28
372.8	0.61	1.3	2.75	5.63	9.85	14.22	18.21	21.65	24.08	25.61	26.59	27.3	27.87	28.36
396.1	0	0	0	0.55	2.81	7.05	12.58	17.66	21.59	23.8	25.02	25.83	26.45	26.98
419.4	0	0	0	0	0	0	1.43	9.95	17.09	20.99	22.76	23.76	24.47	25.07
442.7	0	0	0	0	0	0	0	0	3.75	16.29	19.75	21.15	22	22.67
466	0	0	0	0	0	0	0	0	0	0	0	17.12	18.34	19.28

PART 2: NPSHS Generated Hydrostatic Tables

Draft	Volume	LCB	KB	BM	WL Area	LCF	BML
0	0	0	0	29914.82	900.778	75.17957	1.28E+08
2	7434.564	25.77108	1.252559	47.2335	6533.786	18.9594	5477.666
4	22928.6	19.47325	2.468671	34.86621	8960.248	14.62248	2536.834
6	42759.23	16.3619	3.657526	29.64999	10870.38	11.23298	1788.641
8	66221.85	13.81732	4.850444	26.54098	12592.25	7.407675	1465.498
10	92995.89	11.38217	6.050821	24.3983	14181.78	3.540205	1278.662
12	122860.9	8.937604	7.257941	22.77338	15683.23	-0.67709	1162.4
14	155822.5	6.200525	8.475987	21.39158	17278.35	-7.01928	1121.752
16	191830.7	3.170331	9.703124	20.16323	18729.94	-12.6393	1081.037
18	230773.9	-7.27E-02	10.93662	19.03035	20213.21	-19.2057	1068.772
20	272029.4	-3.03355	12.16051	17.70779	21042.23	-19.9708	974.5967
22	315143.7	-5.57795	13.37091	16.53532	22072.09	-23.2154	933.7149
24	359875.9	-7.71812	14.56835	15.41618	22660.18	-22.3872	854.1186
26	405745.5	-9.31938	15.7481	14.47973	23209.36	-21.3892	787.954
28	452697.3	-10.5143	16.91549	13.7011	23742.47	-20.3039	733.1216

APPENDIX C: NPSHS OUTPUTS FOR INDIVIDUAL DDG-51 COMPARTMENTS

Compartment 1: 0 ft to 42 ft aft FP

233	191						
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	275.4117	202.5389	2566.218	1.22E+07
2	467.4372	206.3237	0.940537	192.0256	203.3627	681.9908	8432887
4	818.6701	205.4192	1.810743	159.2074	204.3626	378.788	6897555
6	1139.169	204.5408	2.708633	161.2917	204.9736	468.2545	6929603
8	1478.622	203.7959	3.697621	178.1609	205.2445	705.2813	7625833
10	1857.753	203.1969	4.783825	200.97	205.3674	1069.809	8587532
12	2285.55	202.7249	5.951108	226.8275	205.4142	1571.871	9686157
14	2767.198	202.3587	7.181381	254.8195	205.4184	2232.965	1.09E+07
16	3308.027	202.0805	8.46279	286.0101	205.384	3107.372	1.22E+07
18	3914.136	201.8776	9.787691	320.0987	205.3193	4229.092	1.37E+07
20	4591.771	201.7382	11.14992	357.5358	205.2401	5689.041	1.53E+07
22	5350.442	201.6717	12.54934	401.1358	205.4183	7546.637	1.72E+07
24	6202.626	201.6955	13.98785	451.0482	205.7733	9906.832	1.94E+07
26	7158.851	201.8031	15.46129	505.1768	206.1286	12884.15	2.18E+07
28	8228.68	201.9752	16.96387	564.652	206.4606	16737.64	2.45E+07

Compartment 2: 42 ft to 78 ft aft FP

191	155						
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	51.7928	182.7631	28.93132	1633909
2	257.5062	172.8288	1.199245	205.7134	170.5394	629.9792	6098629
4	774.2559	170.9787	2.446439	311.0363	169.9941	2276.732	9189457
6	1488.662	170.3857	3.692563	403.3696	169.8236	5043.934	1.19E+07
8	2381.357	170.1049	4.944448	489.3261	169.7596	9054.294	1.44E+07
10	3443.038	169.9538	6.203039	572.3539	169.7623	14473.58	1.69E+07
12	4666.797	169.8729	7.466578	651.4055	169.8052	21232.65	1.92E+07
14	6045.399	169.8345	8.732607	727.1964	169.8668	29342.89	2.15E+07
16	7571.922	169.825	9.999307	799.3268	169.9576	38608.61	2.36E+07
18	9239.627	169.8357	11.26539	868.3781	170.0471	49062.61	2.57E+07
20	11043.75	169.8603	12.53096	935.7495	170.1495	60788.6	2.77E+07
22	12981.69	169.8955	13.79694	1002.185	170.2556	73940.39	2.97E+07
24	15052.63	169.9387	15.06458	1068.761	170.3634	88802.46	3.16E+07
26	17256.62	169.988	16.3348	1135.225	170.4749	105391.6	3.36E+07
28	19595.46	170.0427	17.60892	1203.614	170.5942	124355.1	3.57E+07

Compartment 3: 78 ft to 126 ft aft FP

	155	107					
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	75.86133	126.9185	18.99917	1263787
2	591.4756	127.4736	1.247828	515.6143	127.7273	5677.062	8645705
4	1895.444	127.6306	2.501197	788.3544	127.939	19919.67	1.33E+07
6	3698.192	127.7686	3.739655	1014.393	128.1212	41817.94	1.71E+07
8	5923.647	127.8948	4.975602	1211.063	128.2857	70313.69	2.04E+07
10	8521.65	128.013	6.209404	1386.941	128.4474	104483.3	2.34E+07
12	11452.03	128.1268	7.43979	1543.437	128.6115	142563.3	2.61E+07
14	14678.37	128.2367	8.665104	1682.904	128.7676	183194.1	2.86E+07
16	18167.42	128.3417	9.883981	1806.144	128.9113	224773.3	3.07E+07
18	21891.65	128.4411	11.09627	1918.089	129.042	267508.3	3.26E+07
20	25831.09	128.535	12.30298	2021.345	129.1647	311334.5	3.44E+07
22	29970.63	128.6238	13.50529	2118.199	129.2752	356571.4	3.61E+07
24	34298.58	128.7076	14.70427	2209.751	129.3778	403153.4	3.77E+07
26	38806.83	128.7872	15.9011	2298.498	129.4767	451997.5	3.93E+07
28	43490.43	128.8631	17.09703	2385.107	129.5685	503373.4	4.08E+07

Compartment 4: 126 ft to 174 ft aft FP

	107	59					
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	95.91827	82.97955	31.91875	678955.8
2	1064.182	80.32667	1.273245	968.2637	80.20795	36250.52	6549778
4	3378.314	80.41826	2.493323	1345.869	80.84269	93552.91	9201814
6	6334.112	80.68552	3.676954	1609.929	81.27064	156817.8	1.11E+07
8	9762.651	80.92242	4.851098	1818.611	81.56615	223391.6	1.26E+07
10	13571.53	81.12051	6.019712	1990.27	81.7916	290600.3	1.38E+07
12	17696.4	81.28633	7.18329	2134.594	81.96038	356773.5	1.49E+07
14	22088.3	81.42574	8.341702	2257.314	82.09068	420529.9	1.58E+07
16	26710.23	81.54343	9.495189	2364.61	82.18573	482354.9	1.65E+07
18	31533.84	81.64404	10.64417	2458.996	82.27026	541531.1	1.72E+07
20	36535.78	81.73153	11.78889	2542.947	82.34052	598148.4	1.78E+07
22	41697.15	81.8087	12.92967	2618.422	82.40587	652305.8	1.84E+07
24	47002.18	81.87807	14.06677	2686.611	82.47089	703944.6	1.89E+07
26	52438.42	81.94157	15.20061	2749.631	82.5351	754033.8	1.93E+07
28	57997.13	82.00044	16.33185	2809.077	82.59684	803449.2	1.98E+07

Compartment 5: 174 ft to 220 ft aft FP

59	13						
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	92	36	30.66667	135454.7
2	1460.694	34.91713	1.291344	1368.694	34.86593	102748.4	1941778
4	4556.077	34.96865	2.478391	1726.69	35.13249	204838.7	2471081
6	8235.192	35.06651	3.61407	1952.426	35.27637	295223.5	2807329
8	12310.56	35.14711	4.739589	2122.947	35.37846	378818.4	3062801
10	16690.85	35.21392	5.86036	2257.339	35.45715	454838.6	3265200
12	21313.26	35.27132	6.976727	2365.069	35.52842	522595.7	3429095
14	26130.4	35.32297	8.088227	2452.07	35.59863	581924.8	3563420
16	31106.14	35.37112	9.194603	2523.672	35.66904	633955.4	3675896
18	36213.91	35.41656	10.29607	2584.101	35.73325	680236.1	3771749
20	41435.54	35.45928	11.39335	2637.523	35.79008	723033.6	3856776
22	46758.01	35.49876	12.48722	2684.951	35.83154	762577.8	3931466
24	52172.25	35.53459	13.57848	2729.293	35.86354	800878.7	4000560
26	57671.11	35.5668	14.66774	2769.568	35.88656	836791.9	4062667
28	63248.6	35.59572	15.75544	2807.917	35.90757	871983.2	4121711

Compartment 6: 220 ft to 254 ft aft FP

13	-21						
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	68	-4	22.66667	7638.667
2	1176.528	-3.88826	1.294802	1108.528	-4.11447	98272.91	124354
4	3673.926	-3.93952	2.479367	1388.87	-4.08084	193167	156223.3
6	6621.996	-3.99055	3.610111	1559.199	-4.06607	273312.4	175525.6
8	9862.792	-4.02078	4.728125	1681.597	-4.05529	342863.2	189332.5
10	13318.11	-4.03788	5.838748	1773.72	-4.04697	402343.3	199676.3
12	16935.49	-4.04702	6.942555	1843.66	-4.03984	451821.3	207500.4
14	20674.84	-4.05125	8.038972	1895.687	-4.03304	491140	213291
16	24505.81	-4.05232	9.127722	1935.285	-4.02716	522546	217678.8
18	28407.82	-4.05142	10.2094	1966.722	-4.02161	548412.6	221150
20	32368.73	-4.04942	11.28538	1994.196	-4.01651	571708.9	224186.6
22	36382.91	-4.04681	12.35744	2019.976	-4.01315	594162.3	227034.4
24	40448.08	-4.04386	13.42726	2045.193	-4.01011	616689.4	229829.6
26	44563.65	-4.04094	14.49622	2070.381	-4.00857	639754.8	232640.3
28	48728.85	-4.03821	15.56518	2094.817	-4.00765	662674.6	235370.6

Compartment 7: 254 ft to 300 ft aft FP

	-21	-67					
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	92	-44	30.66667	194334.7
2	1372.629	-42.2415	1.28865	1280.629	-42.9773	86215.59	2588396
4	4387.363	-42.4433	2.499041	1734.105	-43.4546	208768.8	3565696
6	8140.189	-42.7301	3.663699	2018.721	-43.6751	326305.2	4190896
8	12379.42	-42.9556	4.811621	2220.506	-43.796	432538.6	4637423
10	16970.03	-43.1304	5.94757	2370.108	-43.8734	525016.9	4970290
12	21819.74	-43.267	7.072209	2479.604	-43.9172	600732.4	5213524
14	26856.8	-43.3732	8.184948	2557.46	-43.9427	658919.4	5385482
16	32028.21	-43.4558	9.285923	2613.951	-43.9576	703474.3	5509234
18	37299.21	-43.5206	10.37644	2657.048	-43.9664	738809.8	5602920
20	42649.99	-43.5721	11.45862	2693.726	-43.97	769811.4	5681889
22	48071.29	-43.6138	12.53489	2727.572	-43.975	799186.1	5754326
24	53559.17	-43.648	13.60739	2760.312	-43.9757	828305.9	5824093
26	59113.09	-43.6766	14.67796	2793.613	-43.9762	858645.2	5894637
28	64733.34	-43.7007	15.74795	2826.635	-43.9787	889451.4	5964863

Compartment 8: 300 ft to 338 ft aft FP

	-67	-105					
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	75.88048	-85.9887	25.21502	570066.2
2	716.0237	-83.0457	1.262684	640.1432	-83.0956	17879.27	4633161
4	2364.836	-83.1732	2.525921	1008.669	-83.8932	65021.64	7371683
6	4677.954	-83.5564	3.77036	1304.449	-84.4939	134601.3	9608064
8	7527.236	-83.9189	5.00352	1544.834	-84.9278	218321.3	1.14E+07
10	10817.18	-84.2328	6.225184	1745.112	-85.2759	310500.1	1.30E+07
12	14465.83	-84.5013	7.433165	1903.539	-85.5439	400097.3	1.42E+07
14	18384.83	-84.7249	8.621846	2015.453	-85.7143	473451	1.51E+07
16	22490.15	-84.9039	9.78721	2089.875	-85.8052	527267.7	1.57E+07
18	26721.97	-85.0447	10.93011	2141.937	-85.8565	567414.3	1.61E+07
20	31045.85	-85.1556	12.05447	2181.941	-85.8849	599690.3	1.64E+07
22	35442.89	-85.2442	13.16456	2215.105	-85.902	627393.4	1.66E+07
24	39904.2	-85.316	14.26443	2246.209	-85.913	654162.2	1.69E+07
26	44425.98	-85.3752	15.35734	2275.562	-85.9193	680125	1.71E+07
28	49005.8	-85.4246	16.44559	2304.267	-85.9228	706179.2	1.73E+07

Compartment 9: 338 ft to 370 ft aft FP

	-105	-137					
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	56.16002	-121.502	15.67294	817582.1
2	271.9445	-116.809	1.195658	215.7845	-120.403	1229.289	3067178
4	904.1364	-116.643	2.531258	416.4074	-120.881	8089.915	5951544
6	1976.836	-117.044	3.911332	656.292	-121.088	27980.62	9455176
8	3540.864	-117.639	5.299293	907.7362	-121.086	66921.55	1.32E+07
10	5592.509	-118.224	6.670996	1143.909	-121.056	127298.1	1.67E+07
12	8091.101	-118.711	8.016508	1354.683	-121.049	206554.7	1.98E+07
14	10967.99	-119.102	9.328764	1522.211	-121.034	289527.6	2.23E+07
16	14127.47	-119.411	10.5998	1637.265	-121.023	358497.9	2.40E+07
18	17475.64	-119.648	11.82742	1710.903	-121.015	408342.5	2.51E+07
20	20948.06	-119.828	13.01718	1761.514	-121.014	445365	2.59E+07
22	24509.41	-119.966	14.17765	1799.838	-121.013	474917	2.65E+07
24	28141.09	-120.074	15.31657	1831.84	-121.012	500615.3	2.69E+07
26	31833.54	-120.161	16.44008	1860.619	-121.012	524529.1	2.74E+07
28	35580.99	-120.232	17.55251	1886.827	-121.011	546969.9	2.78E+07

Compartment 10: 370 ft to 410 ft aft FP

	-137	-177					
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy
0	0	0	0	17.75349	-139.511	2.236367	394467
2	56.14449	-139.486	1.122526	38.391	-139.475	22.88841	851311.6
4	175.5751	-139.481	2.4806	81.0396	-139.48	214.7428	1797554
6	444.841	-140.834	4.085932	188.2263	-143.015	1791.993	4261509
8	1039.883	-143.59	5.823489	406.8154	-147.821	9681.171	9457547
10	2161.019	-146.438	7.518894	714.3204	-151.389	32481.02	1.69E+07
12	3953.262	-148.814	9.12774	1077.923	-153.816	84256.59	2.59E+07
14	6496.285	-150.782	10.66343	1465.099	-155.346	176814.2	3.57E+07
16	9723.88	-152.298	12.11304	1762.495	-156.15	290985.5	4.33E+07
18	13419.87	-153.339	13.46321	1933.492	-156.449	379469.9	4.77E+07
20	17382	-154.026	14.72712	2028.636	-156.549	437066.9	5.01E+07
22	21502.71	-154.491	15.93022	2092.075	-156.592	478901.1	5.17E+07
24	25735.73	-154.82	17.09369	2140.946	-156.615	513005.6	5.29E+07
26	30059.43	-155.065	18.23138	2182.753	-156.634	543456.6	5.39E+07
28	34463.6	-155.254	19.35232	2221.42	-156.655	572650.9	5.49E+07

Compartment 11: 410 ft to 442 ft aft FP

-177	-209							
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy	
0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
6	2.085751		-177	5.333333	2.085751	-177	3.42E-02	68876.2
8	14.82779		-177	6.958227	10.65628	-177	4.565001	351894.5
10	52.2196		-177	8.522876	26.73553	-177	72.09253	882867.6
12	181.4119	-179.542	10.42609	102.4568	-188.416	684.0118	3586603	
14	731.7929	-182.646	12.51929	447.9242	-195.169	17488.1	1.62E+07	
16	1998.116	-184.749	14.15326	818.3986	-195.025	74953.08	3.00E+07	
18	4067.502	-187.282	15.63702	1250.987	-193.407	165991.5	4.65E+07	
20	6717.718	-189.09	16.97111	1399.228	-193.213	226158.7	5.21E+07	
22	9588.222	-190.027	18.17978	1471.276	-193.171	261761.9	5.48E+07	
24	12578.94	-190.573	19.32709	1519.444	-193.156	287873.7	5.66E+07	
26	15657.5	-190.926	20.44334	1559.113	-193.148	310744.3	5.81E+07	
28	18811.83	-191.173	21.54339	1595.219	-193.142	332626.2	5.94E+07	

Compartment 12: 442 ft to 466 ft aft FP

-209	-233							
Draft	Volume	LCB	KB	Area	LCF	ixx	iyy	
0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
12	3.01E-02		-209	11.33333	3.01E-02	-209	1.85E-05	1318.066
14	0.269406		-209	13.03565	0.209258	-209	6.23E-03	9171.253
16	93.38425	-209.675	15.32521	92.90559	-231.591	438.5617	4527355	
18	588.7519	-209.679	16.90962	402.4621	-231.654	35617.38	1.96E+07	
20	1479.103	-209.68	18.18718	487.8885	-231.657	63452.25	2.38E+07	
22	2888.347	-212.9	19.6096	921.3559	-222.151	116890.5	4.50E+07	
24	4780.477	-215.656	20.95498	970.774	-221.992	135604.5	4.74E+07	
26	6760.475	-216.93	22.14157	1009.224	-221.884	151612.3	4.93E+07	
28	8812.613	-217.669	23.2742	1042.915	-221.802	166747.2	5.10E+07	

APPENDIX D: NPSHS FORM CODE

Code for frmEntry1.frm

```
Private iEdit As Boolean
Private Sub OpenDataFile()
Dim Dummy As Variant
On Error GoTo ErrHandler
dlgOpen.CancelError = True
With dlgOpen
    .Filter = "All Files (*,*)|*.*|Comma Separated Value(*.csv)|*.csv|"
    .DefaultExt = "csv"
    .FilterIndex = 2
    .InitDir = App.Path
    .DialogTitle = "Open file ..."
    .ShowOpen
    InFileName = .FileName
End With
intFile = FreeFile()
Open InFileName For Input As #intFile
Input #intFile, intWL, intST
If iEdit = True Then
    txtST.Text = intST
    txtWL.Text = intWL
    If intST = 0 Or intWL = 0 Then GoTo ErrHandler
End If
Select Case intWL
Case 1
    Input #intFile, Dummy, z(1)
    For j = 1 To intST
    Input #2, X(j), Y(j, 1)
    Next j
Case 2
    Input #intFile, Dummy, z(1), z(2)
    For j = 1 To intST
    Input #2, X(j), Y(j, 1), Y(j, 2)
    Next j
Case 3
    Input #intFile, Dummy, z(1), z(2), z(3)
    For j = 1 To intST
    Input #2, X(j), Y(j, 1), Y(j, 2), Y(j, 3)
    Next j
Case 4
    Input #intFile, Dummy, z(1), z(2), z(3), z(4)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4)
    Next j
Case 5
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5)
    Next j
Case 6
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6)
    For j = 1 To intST
```

```
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6)
    Next j
Case 7
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7)
    Next j
Case 8
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8)
    Next j
Case 9
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9)
    Next j
Case 10
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10)
    Next j
Case 11
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11)
    Next j
Case 12
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12)
    Next j
Case 13
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13)
    Next j
Case 14
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14)
    For j = 1 To intST
```

```
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14)
    Next j
Case 15
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15)
    Next j
Case 16
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16)
    Next j
Case 17
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17)
    Next j
Case 18
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18)
    Next j
Case 19
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19)
    Next j
Case 20
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20)
    Next j
Case 21
```

```
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21)
    Next j
Case 22
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22)
    Next j
Case 23
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23)
    Next j
Case 24
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23), z(24)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23), Y(j, 24)

    Next j
Case 25
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23), z(24), z(25)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23), Y(j, 24), Y(j, 25)
    Next j
Case 26
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23), z(24), z(25), z(26)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
```

```
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23), Y(j, 24), Y(j, 25), Y(j, 26)
    Next j
Case 27
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23), z(24), z(25), z(26), z(27)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23), Y(j, 24), Y(j, 25), Y(j, 26), Y(j,
27)
    Next j
Case 28
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23), z(24), z(25), z(26), z(27),
z(28)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23), Y(j, 24), Y(j, 25), Y(j, 26), Y(j,
27), Y(j, 28)
    Next j
Case 29
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23), z(24), z(25), z(26), z(27),
z(28), z(29)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23), Y(j, 24), Y(j, 25), Y(j, 26), Y(j,
27), Y(j, 28), Y(j, 29)
    Next j
Case 30
    Input #intFile, Dummy, z(1), z(2), z(3), z(4), z(5), z(6), z(7),
z(8), z(9), z(10), z(11), z(12), z(13), z(14), z(15), z(16), z(17),
z(18), z(19), z(20), z(21), z(22), z(23), z(24), z(25), z(26), z(27),
z(28), z(29), z(30)
    For j = 1 To intST
    Input #intFile, X(j), Y(j, 1), Y(j, 2), Y(j, 3), Y(j, 4), Y(j, 5),
Y(j, 6), Y(j, 7), Y(j, 8), Y(j, 9), Y(j, 10), Y(j, 11), Y(j, 12), Y(j,
13), Y(j, 14), Y(j, 15), Y(j, 16), Y(j, 17), Y(j, 18), Y(j, 19), Y(j,
20), Y(j, 21), Y(j, 22), Y(j, 23), Y(j, 24), Y(j, 25), Y(j, 26), Y(j,
27), Y(j, 28), Y(j, 29), Y(j, 30)
    Next j
End Select
Close #intFile
If iEdit = False Then
    Load frmHydro
    frmHydro.Visible = False
    frmHydro.Show
    Unload Me
```

```
    frmHydro.Visible = True
Else
End If
Exit Sub
ErrHandler:
Close #intFile
If Err.Number = 32755 Then
    Exit Sub
Else
    If iEdit = False Then
        MsgBox "Error in opening file " & InFileName & "." & vbCrLf &
"Use EDIT to correct."
    Else
        Prompt = "Error in opening file " & InFileName & "." & vbCrLf _
            & "Do you want to use NOTEPAD to edit?" & vbCrLf _
            & "If you click YES you must save the file and then click
EDIT again."
        Result% = MsgBox(Prompt, vbExclamation + vbYesNo, "File error")
        entry = "start.exe notepad.exe " & InFileName
        If Result% = vbYes Then Result1 = Shell(entry, vbHide)
    End If
    Exit Sub
End If
End Sub


Private Sub cmdEdit_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
iEdit = True
If Button = vbRightButton Then
    entry = "start.exe notepad.exe "
    Result1 = Shell(entry, vbHide)
Else
    OpenDataFile
End If
End Sub

Private Sub cmdEntry2_Click()

If IsNumeric(txtST.Text) = False Then
    MsgBox "Invalid number of stations, please re-enter", vbCritical,
"Input Error!"
    Exit Sub
End If

If IsNumeric(txtWL.Text) = False Then
    MsgBox "Invalid number of waterlines, please re-enter", vbCritical,
"Input Error!"
    Exit Sub
End If

intST = CInt(txtST.Text)
intWL = CInt(txtWL.Text)

If intWL < 2 Then
    MsgBox "You must use at least 2 waterlines", vbCritical, "Input
Error!"
```

```
    txtWL.Text = 2
    intWL = CInt(txtWL.Text)
    Exit Sub
End If
If intST < 2 Then
    MsgBox "You must use at least 2 stations", vbCritical, "Input
Error!"
    txtST.Text = 2
    intST = CInt(txtST.Text)
    Exit Sub
End If

If intST > 41 Then
    MsgBox "Number of Stations cannot exceed 41", vbCritical, "Input
Error!"
    txtST.Text = 41
    intST = CInt(txtST.Text)
Else
    If intWL > 30 Then
        MsgBox "Number of Waterlines cannot exceed 30", vbCritical,
"Input Error!"
        txtWL.Text = 30
        intWL = CInt(txtWL.Text)
    Else
        Me.Hide
        Load frmEntry2
        frmEntry2.Show
    End If
End If

End Sub

Private Sub cmdExit_Click()
End
End Sub

Private Sub cmdOpen_Click()
iEdit = False
OpenDataFile
End Sub

Private Sub Form_Load()
txtST.Text = 11
txtWL.Text = 5
iEdit = False
End Sub

Private Sub Form_Terminate()
Unload Me
End Sub

Private Sub Form_Unload(Cancel As Integer)
Unload Me
End Sub

Private Sub imgHelp_Click()
MsgBox "Help not yet available...", vbOKOnly, "Help Topics"
```

```
End Sub

Private Sub txtST_GotFocus()
txtST.SelStart = 0
txtST.SelLength = Len(txtST.Text)
End Sub

Private Sub txtST_KeyUp(KeyCode As Integer, Shift As Integer)
If KeyCode = 13 Then txtWL.SetFocus
End Sub

Private Sub txtWL_GotFocus()
txtWL.SelStart = 0
txtWL.SelLength = Len(txtWL.Text)
End Sub

Private Sub txtWL_KeyUp(KeyCode As Integer, Shift As Integer)
If KeyCode = 13 Then cmdEntry2.SetFocus
End Sub
```

Code for frmEntry2.frm

```
Private Sub cmdCancel_Click()
Unload Me
frmEntry1.Show
End Sub

Private Sub cmdEntry1_Click()
Unload Me
frmEntry1.Show
End Sub

Private Sub cmdEntry3_Click()
frmEntry3.cmdPrev.Enabled = False
For i = 0 To (intWL - 1)
    z(i + 1) = CSng(txtWL(i).Text)
Next
Me.Hide
Load frmEntry3
frmEntry3.Show
End Sub

Private Sub Form_Load()
On Error GoTo ErrHandler
Me.Caption = "Waterline Data"
For i = 0 To (intWL - 1)
    lblWL(i).Visible = True
    txtWL(i).Visible = True
    lblWL(i).Caption = (i + 1)
    If InFileName = "" Then
        txtWL(i).Text = 0
    Else
        txtWL(i).Text = z(i + 1)
    End If
Next
If intWL <= 15 Then
    Me.Height = txtWL(intWL - 1).Top + txtWL(intWL - 1).Height + 500
    If Me.Height < 2500 Then Me.Height = 2500
    cmdEntry3.Top = Me.Height - cmdEntry3.Height - 500
    cmdEntry1.Top = cmdEntry3.Top - cmdEntry3.Height - 100
    cmdCancel.Top = cmdEntry1.Top - cmdCancel.Height - 100
Else
    Me.Height = txtWL(14).Top + txtWL(14).Height + 500
    cmdEntry3.Top = Me.Height - cmdEntry3.Height - 500
    cmdEntry1.Top = cmdEntry3.Top - cmdEntry3.Height - 100
    cmdCancel.Top = cmdEntry1.Top - cmdCancel.Height - 100
End If
Exit Sub
ErrHandler:
frmEntry1.Show
End Sub

Private Sub Form_Terminate()
Unload frmEntry1
Unload Me
```

```
End
End Sub

Private Sub Form_Unload(Cancel As Integer)
'Unload frmEntry1
Unload Me
End Sub

Private Sub txtWL_GotFocus(Index As Integer)
For i = 0 To (intWL - 1)
    txtWL(i).SelStart = 0
    txtWL(i).SelLength = Len(txtWL(i).Text)
Next
End Sub

Private Sub txtWL_KeyUp(Index As Integer, KeyCode As Integer, Shift As
Integer)
If KeyCode = 13 Then
    If Index < (intWL - 1) Then
        txtWL(Index + 1).SetFocus
    Else
        cmdEntry3.SetFocus
    End If
End If
End Sub
```

Code for frmEntry3.frm

```
Private iStation As Integer
Private OutFileName As String, intFile As Integer

Private Sub cmdCancel_Click()
Unload Me
frmEntry1.Show
End Sub

Private Sub cmdNext_Click()
txtSTLocation.SetFocus
cmdPrev.Enabled = True
X(iStation) = CSng(txtSTLocation.Text)
For i = 1 To intWL
    Y(iStation, i) = CSng(txtY(i - 1).Text)
Next
If cmdNext.Caption = "&Done" Then PrintResults
If iStation = (intST - 1) Then cmdNext.Caption = "&Done"
iStation = iStation + 1
Me.Caption = "Station No. " & iStation & " / " & intST
For i = 0 To (intWL - 1)
    txtY(i).Text = Y(iStation, i + 1)
Next
txtSTLocation.Text = X(iStation)
End Sub

Private Sub cmdPrev_Click()
X(iStation) = CSng(txtSTLocation.Text)
For i = 1 To intWL
    Y(iStation, i) = CSng(txtY(i - 1).Text)
Next
iStation = iStation - 1
cmdNext.Caption = "&Next >>"
Me.Caption = "Station No. " & iStation & " / " & intST
For i = 0 To (intWL - 1)
    txtY(i).Text = Y(iStation, i + 1)
Next
txtSTLocation.Text = X(iStation)
If iStation = 1 Then cmdPrev.Enabled = False
End Sub

Private Sub Form_Load()
iStation = 1
Me.Caption = "Station No. " & iStation & " / " & intST
For i = 1 To intST
    For j = 1 To intWL
        If InFileName = "" Then
            Y(i, j) = 0
        Else
        End If
    Next j
Next i
```

```
For i = 0 To (intWL - 1)
    txtY(i).Text = Y(iStation, i + 1)
Next
If InFileName = "" Then
    txtSTLocation.Text = 0
Else
    txtSTLocation.Text = X(1)
End If
If intWL <= 15 Then
    Me.Height = txtY(intWL - 1).Top + txtY(intWL - 1).Height + 500
    If Me.Height < 2500 Then Me.Height = 2500
    cmdNext.Top = Me.Height - cmdNext.Height - 500
    cmdPrev.Top = cmdNext.Top - cmdPrev.Height - 100
    cmdCancel.Top = cmdPrev.Top - cmdCancel.Height - 100
Else
    Me.Height = txtY(14).Top + txtY(14).Height + 500
    cmdNext.Top = Me.Height - cmdNext.Height - 500
    cmdPrev.Top = cmdNext.Top - cmdPrev.Height - 100
End If
End Sub

Private Sub Form_Resize()
For i = 0 To (intWL - 1)
    lblWL(i).Visible = True
    lblWL(i).Caption = z(i + 1)
    txtY(i).Visible = True
Next i
End Sub

Private Sub Form_Terminate()
Unload frmEntry1
Unload frmEntry2
Unload Me
End Sub

Private Sub Form_Unload(Cancel As Integer)
Unload frmEntry1
Unload frmEntry2
Unload Me
End Sub

Private Sub txtSTLocation_KeyUp(KeyCode As Integer, Shift As Integer)
If KeyCode = 13 Then txtY(0).SetFocus
End Sub

Private Sub txtY_GotFocus(Index As Integer)
For i = 0 To (intWL - 1)
    txtY(i).SelStart = 0
    txtY(i).SelLength = Len(txtY(i).Text)
Next
End Sub
Private Sub txtSTLocation_GotFocus()
txtSTLocation.SelStart = 0
txtSTLocation.SelLength = Len(txtSTLocation.Text)
End Sub
Private Sub PrintSelectCase1()
```

```
PrintString1 = ","
For i = 1 To intWL
    PrintString1 = PrintString1 & z(i) & ","
Next i
Print #intFile, PrintString1
For j = 1 To intST
    PrintString2 = ""
    For i = 1 To intWL
        PrintString2 = PrintString2 & Y(j, i) & ","
    Next i
    Print #intFile, X(j); ","; PrintString2
Next j

End Sub
Private Sub PrintSelectCase()
Select Case intWL
Case 1
    Print #intFile, ""; ","; z(1)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1)
    Next j
Case 2
    Print #intFile, ""; ","; z(1); ","; z(2)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2)
    Next j
Case 3
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3)
    Next j
Case 4
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4)
    Next j
Case 5
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5)
    Next j
Case 6
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6)
    Next j
Case 7
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7)
```

```
    Next j
Case 8
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8)
    Next j
Case 9
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9)
    Next j
Case 10
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10)
    Next j
Case 11
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11)
    Next j
Case 12
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12)
    Next j
Case 13
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13)
    Next j
Case 14
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14)
    For j = 1 To intST
```

```
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14)
    Next j
Case 15
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15)
    Next j
Case 16
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16)

    Next j
Case 17
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17)
    Next j
Case 18
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18)
    Next j
Case 19
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
```

```
", "; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19)
    Next j
Case 20
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20)
    Next j
Case 21
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21)
    Next j
Case 22
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22)
    Next j
Case 23
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23)
    Next j
Case 24
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
```

```
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23); ","; z(24)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23); ","; Y(j, 24)
    Next j
Case 25
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23); ","; z(24); ","; z(25)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23); ","; Y(j, 24); ","; Y(j, 25)
    Next j
Case 26
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23); ","; z(24); ","; z(25); ","; z(26)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23); ","; Y(j, 24); ","; Y(j, 25); ","; Y(j, 26)
    Next j
Case 27
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23); ","; z(24); ","; z(25); ","; z(26); ","; z(27)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23); ","; Y(j, 24); ","; Y(j, 25); ","; Y(j, 26); ","; Y(j,
27)
    Next j
Case 28
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
```

```
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23); ","; z(24); ","; z(25); ","; z(26); ","; z(27); ","; z(28)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23); ","; Y(j, 24); ","; Y(j, 25); ","; Y(j, 26); ","; Y(j,
27); ","; Y(j, 28)
    Next j
Case 29
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23); ","; z(24); ","; z(25); ","; z(26); ","; z(27); ","; z(28); ",";
z(29)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23); ","; Y(j, 24); ","; Y(j, 25); ","; Y(j, 26); ","; Y(j,
27); ","; Y(j, 28); ","; Y(j, 29)
    Next j
Case 30
    Print #intFile, ""; ","; z(1); ","; z(2); ","; z(3); ","; z(4);
","; z(5); ","; z(6); ","; z(7); ","; z(8); ","; z(9); ","; z(10); ",";
z(11); ","; z(12); ","; z(13); ","; z(14); ","; z(15); ","; z(16); ",";
z(17); ","; z(18); ","; z(19); ","; z(20); ","; z(21); ","; z(22); ",";
z(23); ","; z(24); ","; z(25); ","; z(26); ","; z(27); ","; z(28); ",";
z(29); ","; z(30)
    For j = 1 To intST
    Print #intFile, X(j); ","; Y(j, 1); ","; Y(j, 2); ","; Y(j, 3);
","; Y(j, 4); ","; Y(j, 5); ","; Y(j, 6); ","; Y(j, 7); ","; Y(j, 8);
","; Y(j, 9); ","; Y(j, 10); ","; Y(j, 11); ","; Y(j, 12); ","; Y(j,
13); ","; Y(j, 14); ","; Y(j, 15); ","; Y(j, 16); ","; Y(j, 17); ",";
Y(j, 18); ","; Y(j, 19); ","; Y(j, 20); ","; Y(j, 21); ","; Y(j, 22);
","; Y(j, 23); ","; Y(j, 24); ","; Y(j, 25); ","; Y(j, 26); ","; Y(j,
27); ","; Y(j, 28); ","; Y(j, 29); ","; Y(j, 30)
    Next j
End Select
End Sub
Private Sub PrintResults()
dlgSave.CancelError = True
On Error GoTo ErrHandler
dlgSave.Filter = "All Files (*,*)|*.*|Comma Separated
Value(*.csv)|*.csv|"
dlgSave.DefaultExt = "csv"
dlgSave.FilterIndex = 2
dlgSave.InitDir = App.Path
dlgSave.DialogTitle = "Save results as ..."
dlgSave.ShowSave
```

```
OutFileName = dlgSave.FileName
success% = FileExists%(OutFileName)

If success% = False Then
    intFile = FreeFile()
    Open OutFileName For Output As #intFile
    Print #intFile, intWL; ","; intST
    PrintSelectCase
    Close #intFile
    frmHydro.Visible = False
    Load frmHydro
    Unload Me
    Unload frmEntry2
    Unload frmEntry1
    frmHydro.Visible = True
Else
    Prompt = OutFileName & " already exists." & vbCrLf & "Do you want
to replace it?"
    Result% = MsgBox(Prompt, vbExclamation + vbYesNo, "Save file as
...")
    If Result% = vbYes Then
        intFile = FreeFile()
        Open OutFileName For Output As #intFile
        Print #intFile, intWL; ","; intST
        PrintSelectCase
        Close #intFile
        frmHydro.Visible = False
        Load frmHydro
        Unload Me
        Unload frmEntry2
        Unload frmEntry1
        frmHydro.Visible = True
    Else
    End If
End If

ErrHandler:
iStation = iStation - 1
Exit Sub
End Sub
Private Function FileExists%(fname$)
On Error Resume Next
Dim ff%
ff% = FreeFile()
Open fname$ For Input As ff%
If Err Then
    FileExists% = False
Else
    FileExists% = True
End If
Close ff%
End Function

Private Sub txtY_KeyUp(Index As Integer, KeyCode As Integer, Shift As
Integer)
If KeyCode = 13 Then
    If Index < (intWL - 1) Then
```

```
            txtY(Index + 1).SetFocus
        Else
            cmdNext_Click
        End If
    End If
End Sub
```

Code for frmHydro.frm

```
Private iComp As Integer

Private Sub chkLongBlk_Click()
If chkLongBlk.Value = 0 Then
    txtLongBlk.Enabled = False
    spnLongBlk.Enabled = False
    cmbLongBlk.Enabled = False
    lblSelectLongBlk.Visible = False
Else
    txtLongBlk.Enabled = True
    spnLongBlk.Enabled = True
    cmbLongBlk.Enabled = True
    lblSelectLongBlk.Visible = True
End If
End Sub

Private Sub chkShow_Click()
If chkShow.Value = 0 Then
    fraMessage.Visible = False
Else
    fraMessage.Visible = True
End If
End Sub


Private Sub cmdComp_Click()
Dim CompVolume(1 To 30) As Single
Dim CompKB(1 To 30) As Single, CompLCB(1 To 30) As Single
Dim CompWPArea(1 To 30) As Single, CompLCF(1 To 30) As Single,
CompIxx(1 To 30) As Single
Dim CompIyy(1 To 30) As Single, CompBM(1 To 30) As Single, CompBML(1 To
30) As Single
Dim CompFileName As String, YAft(1 To 30) As Single, YFore(1 To 30) As
Single
Dim STAreaAft(1 To 30) As Single, KBStationAft(1 To 30) As Single
Dim STAreaFore(1 To 30) As Single, KBStationFore(1 To 30) As Single

On Error Resume Next

' Check user input

If IsNumeric(txtForBlk.Text) = False Then
    MsgBox "Invalid input, please re-enter", vbCritical, "Input Error!"
    Exit Sub
End If
If IsNumeric(txtAftBlk.Text) = False Then
    MsgBox "Invalid input, please re-enter", vbCritical, "Input Error!"
    Exit Sub
End If

' Read-in compartment locations and change so that 0 is at amidships
```

```
LBP = Abs(X(1) - X(intST))
xF = 0.5 * LBP - CSng(txtForBlk.Text)
xA = 0.5 * LBP - CSng(txtAftBlk.Text)

If xF <= xA Then
    MsgBox "Forward bulkhead must be in front of aft bulkhead!", _
            vbCritical + vbOKOnly, "Input Error!"
    Exit Sub
Else
End If
If xF > X(1) Then
    MsgBox "Forward bulkhead cannot be forward of the FP!", _
            vbCritical + vbOKOnly, "Input Error!"
    Exit Sub
Else
End If
If xA < X(intST) Then
    MsgBox "Aft bulkhead cannot be aft of the AP!", _
            vbCritical + vbOKOnly, "Input Error!"
    Exit Sub
Else
End If

' Locate compartments

For iST = 1 To intST                ' Locate fore compartment
    If (xF - X(iST)) > 0 Then
        iFaft = iST
        iFfore = iST - 1
        GoTo EndForeSearch
    Else
    End If
Next iST
' If loop ends here, then fore compartment could not be located! Alert
user and exit.
MsgBox "Foreward compartment could not be located!", vbCritical +
vbOKOnly, "Unexpected Program Error!"
Exit Sub

EndForeSearch:                      ' Fore compartment has been located

For iST = 1 To intST                ' Locate aft compartment
    If (xA - X(iST)) >= 0 Then
        iAaft = iST
        iAfore = iST - 1
        GoTo EndAftSearch
    Else
    End If
Next iST
' If loop ends here, then aft compartment could not be located! Alert
user and exit.
MsgBox "Aft compartment could not be located!", vbCritical + vbOKOnly,
"Unexpected Program Error!"
Exit Sub

EndAftSearch:                       ' Aft compartment has been located
```

```
' Adjust for Longitudial Bulkheads if they exist

If intCompType = 0 Then      ' No longitudinal bulkheads
Else
    Select Case intCompType
    Case 1   ' Port side compartment
    Case 2   ' Center compartment
    Case 3   ' Starboard side compartment
    End Select
End If

' Calculate offsets for compartment end-stations by linear
interpolation

For iWL = 1 To intWL
    acoef = (Yc(iAaft, iWL) - Yc(iAfore, iWL)) / (X(iAaft) - X(iAfore))
    bcoef = (Yc(iAfore, iWL) * X(iAaft) - Yc(iAaft, iWL) * X(iAfore)) /
(X(iAaft) - X(iAfore))
    YAft(iWL) = acoef * xA + bcoef
    acoef = (Yc(iFaft, iWL) - Yc(iFfore, iWL)) / (X(iFaft) - X(iFfore))
    bcoef = (Yc(iFfore, iWL) * X(iFaft) - Yc(iFaft, iWL) * X(iFfore)) /
(X(iFaft) - X(iFfore))
    YFore(iWL) = acoef * xF + bcoef
Next iWL

' Calculate Sectional Area Properties for the compartment end-stations

' Aft Station

    For iWL = 1 To intWL
        Value = 0
        Value1 = 0
        For j = 1 To (iWL - 1)
            Value = Value + 0.5 * (YAft(j) + YAft(j + 1)) * (z(j + 1) -
z(j))
            Value1 = Value1 + 0.5 * (z(j) + z(j + 1)) * (z(j + 1) -
z(j)) * YAft(j) + (z(j) + 2 * z(j + 1)) * (z(j + 1) - z(j)) * (YAft(j +
1) - YAft(j)) / 6
        Next j
        STAreaAft(iWL) = Value
        If Value = 0 Then
            KBStationAft(iWL) = Value1 / (Value + 0.1)
        Else
            KBStationAft(iWL) = Value1 / Value
        End If
    Next iWL

' Fore Station

    For iWL = 1 To intWL
        Value = 0
        Value1 = 0
        For j = 1 To (iWL - 1)
            Value = Value + 0.5 * (YFore(j) + YFore(j + 1)) * (z(j + 1)
- z(j))
```

```
            Value1 = Value1 + 0.5 * (z(j) + z(j + 1)) * (z(j + 1) -
z(j)) * YFore(j) + (z(j) + 2 * z(j + 1)) * (z(j + 1) - z(j)) * (YFore(j
+ 1) - YFore(j)) / 6
        Next j
        STAreaFore(iWL) = Value
        If Value = 0 Then
            KBStationFore(iWL) = Value1 / (Value + 0.1)
        Else
            KBStationFore(iWL) = Value1 / Value
        End If
    Next iWL

' Single compartment calculations

For iWL = 1 To intWL          ' Loop over waterlines
    ' Initializations
    Value = 0
    Value1 = 0
    Value2 = 0
    Valuf = 0
    Valuf1 = 0
    Valuf2 = 0
    Valuf3 = 0
    ' Calculate properties between aft station of forward compartment
    ' and forward station of aft compartment
    If iAfore > iFaft Then
        For iST = iFaft To (iAfore - 1) ' Loop between existing
stations
            Value = Value + 0.5 * (STArea(iST, iWL) + STArea(iST + 1,
iWL)) * (X(iST) - X(iST + 1))
            Value1 = Value1 + 0.5 * (X(iST) * STArea(iST, iWL) + X(iST
+ 1) * STArea(iST + 1, iWL)) * (X(iST) - X(iST + 1))
            Value2 = Value2 + 0.5 * (KBStation(iST, iWL) * STArea(iST,
iWL) + KBStation(iST + 1, iWL) * STArea(iST + 1, iWL)) * (X(iST) -
X(iST + 1))
            Valuf = Valuf + 0.5 * (Y(iST, iWL) + Y(iST + 1, iWL)) *
(X(iST) - X(iST + 1))
            Valuf1 = Valuf1 + 0.5 * (X(iST) * Y(iST, iWL) + X(iST + 1)
* Y(iST + 1, iWL)) * (X(iST) - X(iST + 1))
            Valuf2 = Valuf2 + 0.5 * (Y(iST, iWL) ^ 3 + Y(iST + 1, iWL)
^ 3) * (X(iST) - X(iST + 1)) / 12
            Valuf3 = Valuf3 + 0.5 * (Y(iST, iWL) + Y(iST + 1, iWL)) *
(X(iST) - X(iST + 1)) _
                * (0.5 * (X(iST) + X(iST + 1))) ^ 2 _
                + (0.5 / 12) * (Y(iST, iWL) + Y(iST + 1, iWL)) *
(Abs(X(iST) - X(iST + 1))) ^ 3
        Next iST
    Else
        Value = 0
        Value1 = 0
        Value2 = 0
        Valuf = 0
        Valuf1 = 0
        Valuf2 = 0
        Valuf3 = 0
    End If
    ' Add contribution between forward compartment and its aft station
```

```
    Value = Value + 0.5 * (STAreaFore(iWL) + STArea(iFaft, iWL)) * (xF
- X(iFaft))
    Value1 = Value1 + 0.5 * (xF * STAreaFore(iWL) + X(iFaft) *
STArea(iFaft, iWL)) * (xF - X(iFaft))
    Value2 = Value2 + 0.5 * (KBStationFore(iWL) * STAreaFore(iWL) +
KBStation(iFaft, iWL) * STArea(iFaft, iWL)) * (xF - X(iFaft))
    Valuf = Valuf + 0.5 * (YFore(iWL) + Yc(iFaft, iWL)) * (xF -
X(iFaft))
    Valuf1 = Valuf1 + 0.5 * (xF * YFore(iWL) + X(iFaft) * Yc(iFaft,
iWL)) * (xF - X(iFaft))
    Valuf2 = Valuf2 + 0.5 * (YFore(iWL) ^ 3 + Yc(iFaft, iWL) ^ 3) * (xF
- X(iFaft)) / 12
    Valuf3 = Valuf3 + 0.5 * (YFore(iWL) + Yc(iFaft, iWL)) * (xF -
X(iFaft)) _
        * (0.5 * (xF + X(iFaft))) ^ 2 _
        + (0.5 / 12) * (YFore(iWL) + Yc(iFaft, iWL)) * (Abs(xF -
X(iFaft))) ^ 3

    ' Add contribution between aft compartment and its foreward station
    Value = Value + 0.5 * (STArea(iAfore, iWL) + STAreaAft(iWL)) *
(X(iAfore) - xA)
    Value1 = Value1 + 0.5 * (X(iAfore) * STArea(iAfore, iWL) + xA *
STAreaAft(iWL)) * (X(iAfore) - xA)
    Value2 = Value2 + 0.5 * (KBStation(iAfore, iWL) * STArea(iAfore,
iWL) + KBStationAft(iWL) * STAreaAft(iWL)) * (X(iAfore) - xA)
    Valuf = Valuf + 0.5 * (YAft(iWL) + Yc(iAfore, iWL)) * (X(iAfore) -
xA)
    Valuf1 = Valuf1 + 0.5 * (X(iAfore) * YAft(iWL) + xA * Yc(iAfore,
iWL)) * (X(iAfore) - xA)
    Valuf2 = Valuf2 + 0.5 * (YAft(iWL) ^ 3 + Yc(iAfore, iWL) ^ 3) *
(X(iAfore) - xA) / 12
    Valuf3 = Valuf3 + 0.5 * (YAft(iWL) + Yc(iAfore, iWL)) * (X(iAfore)
- xA) _
        * (0.5 * (X(iAfore) + xA)) ^ 2 _
        + (0.5 / 12) * (YAft(iWL) + Yc(iAfore, iWL)) * (Abs(X(iAfore) -
xA)) ^ 3

    CompVolume(iWL) = Value
    If Value = 0 Then
        CompLCB(iWL) = Value1 / (Value + 0.1)
        CompKB(iWL) = Value2 / (Value + 0.1)
    Else
        CompLCB(iWL) = Value1 / Value
        CompKB(iWL) = Value2 / Value
    End If
    CompWPArea(iWL) = Valuf
    If Valuf = 0 Then
        CompLCF(iWL) = Valuf1 / (Valuf + 0.1)
        CompIxx(iWL) = Valuf2
        CompIyy(iWL) = Valuf3
    Else
        CompLCF(iWL) = Valuf1 / Valuf
        CompIxx(iWL) = Valuf2
        CompIyy(iWL) = Valuf3
    End If
    If CompVolume(iWL) = 0 Then
        CompBM(iWL) = CompIxx(iWL) / (CompVolume(iWL) + 0.1)
```

```
        CompBML(iWL) = (CompIyy(iWL) - CompWPArea(iWL) * CompLCF(iWL) ^
2) / (CompVolume(iWL) + 0.1)
    Else
        CompBM(iWL) = CompIxx(iWL) / CompVolume(iWL)
        CompBML(iWL) = (CompIyy(iWL) - CompWPArea(iWL) * CompLCF(iWL) ^
2) / CompVolume(iWL)
    End If

Next iWL

' Export results file

dlgExport.FileName = ""
dlgExport.CancelError = True
With dlgExport
    .Filter = "All Files (*,*)|*.*|Comma Separated Value(*.csv)|*.csv|"
    .DefaultExt = "csv"
    .FilterIndex = 2
    .InitDir = App.Path
    .DialogTitle = "Save compartment results as ..."
    .ShowSave
    CompFileName = .FileName
End With
On Error GoTo ErrHandler
TextMessage = "Hydrostatic calculations for your selected compartment
are complete. " _
    & "Results have been saved in " & CompFileName & ". You may proceed
by selecting another " _
    & "compartment and performing the calculations. You will be " _
    & "prompted for a compartment file name. Plotting and cross curves
of stability calculations " _
    & "are not yet available."

' Check if output file exists

success% = FileExists%(dlgExport.FileName)

If success% = False Then     ' File does not exist
    iComp = FreeFile()
    Open CompFileName For Output As #iComp
    Print #iComp, X(xF); ","; X(xA)
    Print #iComp, "Draft"; ","; "Volume"; ","; "LCB"; ","; "KB"; ",";
"Area"; ","; "LCF"; ","; "ixx"; ","; "iyy"
    For i = 1 To intWL
        Print #iComp, z(i); ","; CompVolume(i); ","; CompLCB(i); ",";
CompKB(i); ","; CompWPArea(i); ","; CompLCF(i); ","; CompIxx(i); ",";
CompIyy(i)
    Next i
    Close #iComp
    txtMessage.Text = TextMessage
Else         ' File already exists
    Prompt = CompFileName & " already exists." & vbCrLf & "Do you want
to replace it?"
    Result% = MsgBox(Prompt, vbExclamation + vbYesNo, "Save file as
...")
    If Result% = vbYes Then     ' Replace existing file
        iComp = FreeFile()
```

```
        Open CompFileName For Output As #iComp
        Print #iComp, xF; ","; xA
        Print #iComp, "Draft"; ","; "Volume"; ","; "LCB"; ","; "KB";
","; "Area"; ","; "LCF"; ","; "ixx"; ","; "iyy"
        For i = 1 To intWL
            Print #iComp, z(i); ","; CompVolume(i); ","; CompLCB(i);
","; CompKB(i); ","; CompWPArea(i); ","; CompLCF(i); ","; CompIxx(i);
","; CompIyy(i)
        Next i
        Close #iComp
        txtMessage.Text = TextMessage
    Else    ' Do not replace existing file
    End If
End If

ErrHandler:     ' If user presses CANCEL then exit subroutine
Exit Sub
End Sub

Private Sub cmdExit_Click()
Unload Me
End
End Sub
Private Sub cmdHydro_Click()
Dim Value As Single, iST As Integer, iWL As Integer, Value1 As Single

' Calculate Sectional Area Curves
For iST = 1 To intST
    For iWL = 1 To intWL
        Value = 0
        Value1 = 0
        For j = 1 To (iWL - 1)
            Value = Value + 0.5 * (Y(iST, j) + Y(iST, j + 1)) * (z(j +
1) - z(j))
            Value1 = Value1 + 0.5 * (z(j) + z(j + 1)) * (z(j + 1) -
z(j)) * Y(iST, j) + (z(j) + 2 * z(j + 1)) * (z(j + 1) - z(j)) * (Y(iST,
j + 1) - Y(iST, j)) / 6
        Next j
        STArea(iST, iWL) = Value
        If Value = 0 Then
            KBStation(iST, iWL) = Value1 / (Value + 0.1)
        Else
            KBStation(iST, iWL) = Value1 / Value
        End If
    Next iWL
Next iST

' Calculate Displacement properties
For iWL = 1 To intWL
    Value = 0
    Value1 = 0
    Value2 = 0
    For iST = 1 To (intST - 1)
        iST1 = iST + 1
        Value = Value + 0.5 * (STArea(iST, iWL) + STArea(iST + 1, iWL))
* (X(iST) - X(iST + 1))
```

```
        Value1 = Value1 + 0.5 * (X(iST) * STArea(iST, iWL) + X(iST + 1)
 * STArea(iST + 1, iWL)) * (X(iST) - X(iST + 1))
        Value2 = Value2 + 0.5 * (KBStation(iST, iWL) * STArea(iST, iWL)
 + KBStation(iST + 1, iWL) * STArea(iST + 1, iWL)) * (X(iST) - X(iST +
 1))
    Next iST
    Volume(iWL) = Value
    If Value = 0 Then
        LCB(iWL) = Value1 / (Value + 0.1)
        KB(iWL) = Value2 / (Value + 0.1)
    Else
        LCB(iWL) = Value1 / Value
        KB(iWL) = Value2 / Value
    End If
Next iWL

' Calculate Waterplane properties
For iWL = 1 To intWL
    Value = 0
    Value1 = 0
    Value2 = 0
    Value3 = 0
    For iST = 1 To (intST - 1)
        Value = Value + 0.5 * (Y(iST, iWL) + Y(iST + 1, iWL)) * (X(iST)
 - X(iST + 1))
        Value1 = Value1 + 0.5 * (X(iST) * Y(iST, iWL) + X(iST + 1) *
 Y(iST + 1, iWL)) * (X(iST) - X(iST + 1))
        Value2 = Value2 + 0.5 * (Y(iST, iWL) ^ 3 + Y(iST + 1, iWL) ^ 3)
 * (X(iST) - X(iST + 1)) / 12
        Value3 = Value3 + 0.5 * (Y(iST, iWL) + Y(iST + 1, iWL)) *
 (X(iST) - X(iST + 1)) _
            * (0.5 * (X(iST) + X(iST + 1))) ^ 2 _
            + (0.5 / 12) * (Y(iST, iWL) + Y(iST + 1, iWL)) *
 (Abs(X(iST) - X(iST + 1))) ^ 3
    Next iST
    WPArea(iWL) = Value
    If Value = 0 Then
        LCF(iWL) = Value1 / (Value + 0.1)
        Ixx(iWL) = Value2
        Iyy(iWL) = Value3
    Else
        LCF(iWL) = Value1 / Value
        Ixx(iWL) = Value2
        Iyy(iWL) = Value3
    End If
    If Volume(iWL) = 0 Then
        BM(iWL) = Ixx(iWL) / (Volume(iWL) + 0.1)
        BML(iWL) = (Iyy(iWL) - WPArea(iWL) * LCF(iWL) ^ 2) /
 (Volume(iWL) + 0.1)
    Else
        BM(iWL) = Ixx(iWL) / Volume(iWL)
        BML(iWL) = (Iyy(iWL) - WPArea(iWL) * LCF(iWL) ^ 2) /
 Volume(iWL)
    End If
Next iWL

HydroExport
```

```
End Sub

Private Sub HydroExport()
Dim OutFileName As String, TextMessage As String
dlgExport.CancelError = True
On Error GoTo ErrHandler
With dlgExport
     .Filter = "All Files (*,*)|*.*|Comma Separated Value(*.csv)|*.csv|"
     .DefaultExt = "csv"
     .FilterIndex = 2
     .InitDir = App.Path
     .DialogTitle = "Save hydrostatic results as ..."
     .ShowSave
OutFileName = .FileName
End With
success% = FileExists%(OutFileName)
TextMessage = "The hydrostatic calculations for your ship data are
complete. " _
     & "Results have been saved in " & OutFileName & ". Plotting of the
results " _
     & "is not yet available. You may proceed by selecting compartments
" _
     & "and performing the calculations. At the end of each calculations
cycle you will be " _
     & "prompted for a compartment file name. You may perform
compartment calculations for " _
     & "practically an unlimited number of compartments."

If success% = False Then
    intFile = FreeFile()
    Open OutFileName For Output As #intFile
    Print #intFile, "Draft"; ","; "Volume"; ","; "LCB"; ","; "KB"; ",";
"BM"; ","; "WL Area"; ","; "LCF"; ","; "BML"
    For i = 1 To intWL
        Print #intFile, z(i); ","; Volume(i); ","; LCB(i); ","; KB(i);
","; BM(i); ","; WPArea(i); ","; LCF(i); ","; BML(i)
    Next i
    Close #intFile
    txtMessage.Text = TextMessage
Else
    Prompt = OutFileName & " already exists." & vbCrLf & "Do you want
to replace it?"
    Result% = MsgBox(Prompt, vbExclamation + vbYesNo, "Save file as
...")
    If Result% = vbYes Then
        intFile = FreeFile()
        Open OutFileName For Output As #intFile
        Print #intFile, "Draft"; ","; "Volume"; ","; "LCB"; ","; "KB";
","; "BM"; ","; "WL Area"; ","; "LCF"; ","; "BML"
        For i = 1 To intWL
            Print #intFile, z(i); ","; Volume(i); ","; LCB(i); ",";
KB(i); ","; BM(i); ","; WPArea(i); ","; LCF(i); ","; BML(i)
        Next i
        Close #intFile
        txtMessage.Text = TextMessage
    Else
```

```
    End If
End If

cmdComp.Enabled = True
cmdHydroPlot.Enabled = False
cmdHydro.Enabled = False
cmdLongBlk.Enabled = True
spnAftBlk.Enabled = True
spnForBlk.Enabled = True
txtAftBlk.Enabled = True
txtForBlk.Enabled = True

PlotFileName = OutFileName

ErrHandler:
Exit Sub
End Sub

Private Sub cmdHydroPlot_MouseUp(Button As Integer, Shift As Integer, X
As Single, Y As Single)
If Button = vbRight Then
    DefaultFile = False
Else
    DefaultFile = True
End If
frmChart.Show
End Sub

Private Sub cmdLongBlk_Click()
Load frmLongBulk
frmLongBulk.Show vbModal
End Sub

Private Sub cmdNew_Click()
Unload Me
Load frmEntry1
frmEntry1.Show
End Sub

Private Sub Form_Load()
Dim LBP As Single
On Error GoTo ErrLogic
Me.Caption = "Hydrostatic Calculations"
cmdHydro.Enabled = True
cmdHydroPlot.Enabled = False
cmdCompPlot.Enabled = False
cmdComp.Enabled = False
cmdStability.Enabled = False
cmdStabilityPlot.Enabled = False
cmdLongBlk.Enabled = False

' Turn half-offsets into full-offsets

For i = 1 To intST
    For j = 1 To intWL
        Y(i, j) = 2 * Y(i, j)
```

```
    Next j
Next i

' Back-up table of offsets

For i = 1 To intST
    For j = 1 To intWL
        Yc(i, j) = Y(i, j)
    Next j
Next i

' Initialize values for compartment calculations

txtForBlk.Text = X(1)
txtAftBlk.Text = X(intST)

' Change Stations so that 0 is at amidships
LBP = Abs(X(1) - X(intST))
For i = 1 To intST
    X(i) = 0.5 * LBP - X(i)
Next i

txtMessage.Text = "You may begin the hydrostatic calculations for your
ship data by pressing the 'Calculate' button. At the end " _
& "of the calculations you will prompted for a file name. All file
names will assume the " _
& "program default extension (.csv) unless specified otherwise. Once
the hydrostatic calculations " _
& "are complete you may start your individual compartment
calculations."

Exit Sub
ErrLogic:
Unload Me
Load frmEntry1
frmEntry1.Show
End Sub


Private Sub spnForBlk_SpinDown()
cmdComp.SetFocus
txtForBlk.Text = txtForBlk - 1
End Sub
Private Sub spnForBlk_SpinUp()
cmdComp.SetFocus
txtForBlk.Text = txtForBlk.Text + 1
End Sub
Private Sub spnAftBlk_SpinDown()
cmdComp.SetFocus
txtAftBlk.Text = txtAftBlk.Text - 1
End Sub
Private Sub spnAftBlk_SpinUp()
cmdComp.SetFocus
txtAftBlk.Text = txtAftBlk.Text + 1
End Sub
Private Sub txtAftBlk_GotFocus()
txtAftBlk.SelStart = 0
```

```
txtAftBlk.SelLength = Len(txtAftBlk.Text)
End Sub

Private Sub txtForBlk_GotFocus()
txtForBlk.SelStart = 0
txtForBlk.SelLength = Len(txtForBlk.Text)
End Sub
Private Function FileExists%(fname$)
On Error Resume Next
Dim ff%
ff% = FreeFile()
Open fname$ For Input As ff%
If Err Then
    FileExists% = False
Else
    FileExists% = True
End If
Close ff%
End Function

Private Sub imgHelp_Click()
MsgBox "Help not yet available...", vbOKOnly, "Help Topics"
End Sub
```

Code for frmLongBulk.frm

```
Private Sub cmdCancel_Click()
intCompType = 0
Unload Me
End Sub

Private Sub cmdOK_Click()
If IsNumeric(txt1.Text) = False Then
    MsgBox "Invalid input for 'Location 1', please re-enter",
vbCritical, "Input Error!"
    Exit Sub
Else
    YBlk1 = CSng(txt1.Text)
End If
If optCenter.Value = True Then
    If IsNumeric(txt2.Text) = False Then
        MsgBox "Invalid input for 'Location 2', please re-enter",
vbCritical, "Input Error!"
        Exit Sub
    Else
        YBlk2 = CSng(txt2.Text)
    End If
End If
MsgBox "Longitudinal bulkhead calculations have not been implemented
yet", vbOKOnly, "Not available!"
Unload Me
End Sub

Private Sub Form_Load()
txt1.Enabled = False
txt2.Enabled = False
txt1.Visible = False
txt2.Visible = False
lbl1.Visible = False
lbl2.Visible = False
cmdOK.Enabled = False
txtMessage.Text = "Port Side:" & vbCrLf & "Compartment located between
'Location 1' and ship's port side." _
    & " Enter a value for 'Location 1' only (positive starboard,
negative port)." & vbCrLf & vbCrLf _
    & "Center Side:" & vbCrLf & "Compartment located between 'Location
1' and 'Location 2'." _
    & " Enter values for both 'Location 1' and 'Location 2' (positive
starboard, negative port)." & vbCrLf & vbCrLf _
    & "Starboard Side:" & vbCrLf & "Compartment located between
'Location 1' and ship's starboard side." _
    & " Enter a value for 'Location 1' only (positive starboard,
negative port)."
intCompType = 0
End Sub

Private Sub optCenter_Click()
txt1.Enabled = True
```

```
txt2.Enabled = True
txt1.Visible = True
txt2.Visible = True
lbl1.Visible = True
lbl2.Visible = True
cmdOK.Enabled = True
txtMessage.Text = "You selected a center compartment." & vbCrLf &
"Enter a value for both 'Location 1' and 'Location 2' " _
    & "(positive starboard, negative port of centerline)."
txt1.SetFocus
intCompType = 2
End Sub

Private Sub optPort_Click()
txt1.Enabled = True
txt2.Enabled = False
txt2.Text = ""
txt1.Visible = True
txt2.Visible = False
lbl1.Visible = True
lbl2.Visible = False
cmdOK.Enabled = True
txtMessage.Text = "You selected a port side compartment." & vbCrLf &
"Enter a value for 'Location 1' " _
    & "(positive starboard, negative port of centerline)."
txt1.SetFocus
intCompType = 1
End Sub

Private Sub optStarboard_Click()
txt1.Enabled = True
txt2.Enabled = False
txt2.Text = ""
txt1.Visible = True
txt2.Visible = False
lbl1.Visible = True
lbl2.Visible = False
cmdOK.Enabled = True
txtMessage.Text = "You selected a starboard side compartment." & vbCrLf
& "Enter a value for 'Location 1' " _
    & "(positive starboard, negative port of centerline)."
txt1.SetFocus
intCompType = 3
End Sub
```

Code for Module1.bas

```
Public intWL As Integer, intST As Integer, intCompType As Integer
Public z(1 To 30) As Single, Y(1 To 41, 1 To 30) As Single, X(1 To 41)
As Single, Yc(1 To 41, 1 To 30) As Single
Public Ixx(1 To 30) As Single, Iyy(1 To 30) As Single, LCF(1 To 30) As
Single
Public STArea(1 To 41, 1 To 30) As Single, WPArea(1 To 30) As Single
Public KBStation(1 To 41, 1 To 30), KB(1 To 30) As Single
Public BM(1 To 30) As Single, LCB(1 To 30) As Single, Volume(1 To 30)
As Single
Public InFileName As String, BML(1 To 30) As Single, PlotFileName As
String
Public DefaultFile As Boolean, xF As Single, xA As Single, YBlk1 As
Single, YBlk2 As Single
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E: EXCEL TRIMMING PROGRAM WORKSHEETS

Table 1: Startup Page

SIMULATION START UP PAGE

When starting a new simulation, follow these steps:

STEP 1: Enter the following variables

Enter the starting KG (ft):	27.60
Enter the LBP (ft):	466.00
Enter the Initial Draft at the Bow (ft):	21.50
Enter the Initial Draft at the Stern (ft):	21.50
Enter the Initial Displacement of the Vessel (LTsw):	[illegible]

STEP 2: Enter initial data into the following cells:
Enter the Initial KG into Cell E3 on the Main page
Enter the Initial draft at the bow into Cell E7 on the Main Page
Enter the Initial draft at the stern into Cell E8 on the Main Page

STEP 3: Ensure that the proper holes are activated in the Excel DDE transfer Macro entitled "Poke"

STEP 4: Zero the iteration counter by entering a "0" into Cell D59 on the Main page

STEP 5: Save the changes that you have made, close Excel and start the simulation in SIMSMART

Table 1: Startup Page

Table 2: Main Page

Main Operating Page

KG (ft): 23.36

LBP (ft): 466.00

Draft at the Bow (ft): 23.97
Draft at the Stern (ft): 25.96

Flooded Water

Compartment —	Volume (ft^3)	VCG (ft)	LCG (ft fwd)
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	32699.02539	11.37359141	-4.049205832
7	30422.03711	8.943974441	-43.43009862
8	20037.5625	9.091001865	-84.79696251
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0

Total Volume of Flooded Water (ft^3) = 83158.63
Overall KG of Flooded Water (ft) = 9.93
Overall LCG of Flooded Water (ft fwd) = -37.91
Overall I_T of Flooded Water (ft^4) = 597121.50
Total Weight of Flooded Water (LTsw) = 2375.96
TPI (LTsw/in) = 54.58
MCT1" (ft-LTsw) = 1584.83
LCF (ft fwd) = -21.91

RESULTS

New Draft at the bow (ft) = 24.02
New Draft at the stern (ft) = 26.04

Compartment	New Pressure at Hole (psi)
1	0.00
2	0.00
3	0.00
4	0.00
5	0.00
6	20.67
7	0.00
8	0.00
9	0.00
10	0.00
11	0.00
12	0.00

Number of updates performed: 40

REMARKS:

Table 2: Main Page

Table 3: Input Page

Data Input Manager

SIMSMART DDE link is into yellow boxes

Inputs

Compartment —	Long'l Location of hole (ft fwd)	Vertical Location of hole (ft above BL)	Volume (ft^3)	Current Pressure at Hole (psi)	KG (ft)	LCG (ft fwd)	I_T (ft^4)	Weight of Floodwater (LTsw)	Weighted KG (ft^4)	Weighted LCG (ft^4)	Weighted I_T (ft^7)	Weighted KG (ft-LTsw)
1	103.00	08.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	13.00	11.50	22689.03	28.62	11.37	-4.05	573556.41	934.26	371905.35	-132405.08	18754735750.94	10625.87
7			30431.04	0.00	8.94	-43.43	689636.13	869.20	272093.92	-1321232.07	20980135810.20	7774.11
8			30037.59	0.00	9.09	-84.80	495116.69	572.50	182161.52	-1699124.44	9920931711.75	5204.61
9			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	-157.00	38.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Summation =	83158.63					Summation =	826160.79	-3152761.59	49655803272.89	23604.59

Overall Volume of Flooded Water (ft^3) =	83158.63
Overall KG of Flooded Water (ft above BL) =	9.93
Overall LCG of Flooded Water (ft fwd) =	-37.91
Overall I_T of Flooded Water (ft^4) =	597121.50

Data Flow

Compartment	Volume	KG	LCG	I_T
2	83158.63	9.93	-37.91	597121.50
3	83158.63	9.93	-37.91	597121.50
4	83158.63	9.93	-37.91	597121.50
5	83158.63	9.93	-37.91	597121.50
6	83158.63	9.93	-37.91	597121.50
7	83158.63	9.93	-37.91	597121.50
8	83158.63	9.93	-37.91	597121.50
9	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00
Overall	83158.63	9.93	-37.91	597121.50

Table 3: Input Page

Table 4: Hydrostatics Page

Hydrostatic Curves for DDG-51
Based on Values Calculated from NPSHS program

Ship Hydrostatic Values for KG = 23.36331492

Actual Values

Draft (ft)	Volume (ft^3)	Displacement (LTsw)	KB (ft)	BM_T (ft)	KM_T (ft)	LCB (ft fwd)	LCF (ft fwd)	BM_L (ft)	A_{WP} (ft^2)	TPI (LTsw/in)	MCT1" (LTsw)
0	0.00	0.00	0.00	29914.82	29914.82	0.00	75.18	1.28E+08	900.78	2.14	0.00
2	7434.56	212.42	1.25	47.23	48.49	25.77	18.96	5477.67	6533.79	15.56	207.23
4	22928.60	655.10	2.47	34.87	37.33	19.47	14.62	2536.83	8960.25	21.33	294.74
6	42759.23	1221.69	3.66	29.65	33.31	16.36	11.23	1788.64	10870.38	25.88	386.46
8	66221.85	1892.05	4.85	26.54	31.39	13.82	7.41	1465.50	12592.25	29.98	489.59
10	92995.89	2657.03	6.05	24.40	30.45	11.38	3.54	1278.66	14181.78	33.77	599.33
12	122860.90	3510.31	7.26	22.77	30.03	8.94	-0.68	1162.40	15683.23	37.34	719.57
14	155822.50	4452.07	8.48	21.39	29.87	6.20	-7.02	1121.75	17278.35	41.14	881.23
16	191830.70	5480.88	9.70	20.16	29.87	3.17	-12.64	1081.04	18729.94	44.60	1046.17
18	230773.90	6593.54	10.94	19.03	29.97	-0.07	-19.21	1068.77	20213.21	48.13	1245.54
20	272029.40	7772.27	12.16	17.71	29.87	-3.03	-19.97	974.60	21042.23	50.10	1339.01
22	315143.70	9004.11	13.37	16.54	29.91	-5.58	-23.22	933.71	22072.09	52.55	1487.36
24	359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
26 ..	405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
28	452697.30	12934.21	16.92	13.70	30.62	-10.51	-20.30	733.12	23742.47	56.53	1680.79

Normalized Values for Curve Fitting

Draft (ft) Multiplier	Volume (ft^3)	Displacement (LTsw) 450	KB (ft) 1	BM_T (ft) 2	KM_T (ft) 2	LCB (ft fwd)	LCF (ft fwd)	BM_L (ft) 200	A_{WP} (ft^2) 800	TPI (LTsw/in) 2	MCT1" (LTsw) 163.6
2	7434.56	0.47	1.25	23.62	24.24	25.77	18.96	27.39	8.17	7.78	1.27
4	22928.60	1.46	2.47	17.43	18.67	19.47	14.62	12.68	11.20	10.67	1.80
6	42759.23	2.71	3.66	14.82	16.65	16.36	11.23	8.94	13.59	12.94	2.36
8	66221.85	4.20	4.85	13.27	15.70	13.82	7.41	7.33	15.74	14.99	2.99
10	92995.89	5.90	6.05	12.20	15.22	11.38	3.54	6.39	17.73	16.88	3.66
12	122860.90	7.80	7.26	11.39	15.02	8.94	-0.68	5.81	19.60	18.67	4.40
14	155822.50	9.89	8.48	10.70	14.93	6.20	-7.02	5.61	21.60	20.57	5.39
16	191830.70	12.18	9.70	10.08	14.93	3.17	-12.64	5.41	23.41	22.30	6.39
18	230773.90	14.65	10.94	9.52	14.98	-0.07	-19.21	5.34	25.27	24.06	7.61
20	272029.40	17.27	12.16	8.85	14.93	-3.03	-19.97	4.87	26.30	25.05	8.18
22	315143.70	20.01	13.37	8.27	14.95	-5.58	-23.22	4.67	27.59	26.28	9.09
24	359875.90	22.85	14.57	7.71	14.99	-7.72	-22.39	4.27	28.33	26.98	9.50
26	405745.50	25.76	15.75	7.24	15.11	-9.32	-21.39	3.94	29.01	27.63	9.89
28	452697.30	28.74	16.92	6.85	15.31	-10.51	-20.30	3.67	29.68	28.26	10.27

Compartment 1 From 0 to 42 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	275.41	202.54	2566.22	12228940.00
2	467.44	206.32	0.94	192.03	203.36	681.99	8432887.00
4	818.67	205.42	1.81	159.21	204.36	378.79	6897555.00
6	1139.17	204.54	2.71	161.29	204.97	468.25	6929603.00
8	1478.62	203.80	3.70	178.16	205.24	705.28	7625833.00
10	1857.75	203.20	4.78	200.97	205.37	1069.81	8587532.00
12	2285.55	202.72	5.95	226.83	205.41	1571.87	9686157.00
14	2767.20	202.36	7.18	254.82	205.42	2232.97	10880860.00
16	3308.03	202.08	8.46	286.01	205.38	3107.37	12218510.00
18	3914.14	201.88	9.79	320.10	205.32	4229.09	13687050.00
20	4591.77	201.74	11.15	357.54	205.24	5689.04	15304560.00
22	5350.44	201.67	12.55	401.14	205.42	7546.64	17213040.00
24	6202.63	201.70	13.99	451.05	205.77	9906.83	19418500.00
26	7158.85	201.80	15.46	505.18	206.13	12884.15	21818120.00
28	8228.68	201.98	16.96	564.65	206.46	16737.64	24459390.00

Compartment 2 From 42 to 78 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	51.79	182.76	28.93	1633909.00
2	257.51	172.83	1.20	205.71	170.54	629.98	6098629.00
4	774.26	170.98	2.45	311.04	169.99	2276.73	9189457.00
6	1488.66	170.39	3.69	403.37	169.82	5043.93	11905410.00
8	2381.36	170.10	4.94	489.33	169.76	9054.29	14438010.00
10	3443.04	169.95	6.20	572.35	169.76	14473.58	16889110.00
12	4666.80	169.87	7.47	651.41	169.81	21232.65	19227100.00
14	6045.40	169.83	8.73	727.20	169.87	29342.89	21472360.00
16	7571.92	169.83	10.00	799.33	169.96	38608.61	23614580.00
18	9239.63	169.84	11.27	868.38	170.05	49062.61	25668000.00
20	11043.75	169.86	12.53	935.75	170.15	60788.60	27675490.00
22	12981.69	169.90	13.80	1002.19	170.26	73940.39	29658160.00
24	15052.63	169.94	15.06	1068.76	170.36	88802.46	31647820.00
26	17256.62	169.99	16.33	1135.23	170.47	105391.60	33636800.00
28	19595.46	170.04	17.61	1203.61	170.59	124355.10	35686810.00

Compartment 3 From 78 to 126 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	75.86	126.92	19.00	1263787.00
2	591.48	127.47	1.25	515.61	127.73	5677.06	8645705.00
4	1895.44	127.63	2.50	788.35	127.94	19919.67	13251860.00
6	3698.19	127.77	3.74	1014.39	128.12	41817.94	17087890.00
8	5923.65	127.89	4.98	1211.06	128.29	70313.69	20439330.00
10	8521.65	128.01	6.21	1386.94	128.45	104483.30	23449750.00
12	11452.03	128.13	7.44	1543.44	128.61	142563.30	26141520.00
14	14678.37	128.24	8.67	1682.90	128.77	183194.10	28550480.00
16	18167.42	128.34	9.88	1806.14	128.91	224773.30	30686680.00
18	21891.65	128.44	11.10	1918.09	129.04	267508.30	32632260.00
20	25831.09	128.54	12.30	2021.35	129.16	311334.50	34431930.00
22	29970.63	128.62	13.51	2118.20	129.28	356571.40	36122640.00
24	34298.58	128.71	14.70	2209.75	129.38	403153.40	37723700.00
26	38806.83	128.79	15.90	2298.50	129.48	451997.50	39278530.00
28	43490.43	128.86	17.10	2385.11	129.57	503373.40	40797250.00

Compartment 4 From 126 to 174 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	95.92	82.98	31.92	678955.80
2	1064.18	80.33	1.27	968.26	80.21	36250.52	6549778.00
4	3378.31	80.42	2.49	1345.87	80.84	93552.91	9201814.00
6	6334.11	80.69	3.68	1609.93	81.27	156817.80	11084620.00
8	9762.65	80.92	4.85	1818.61	81.57	223391.60	12581760.00
10	13571.53	81.12	6.02	1990.27	81.79	290600.30	13819730.00
12	17696.40	81.29	7.18	2134.59	81.96	356773.50	14862600.00
14	22088.30	81.43	8.34	2257.31	82.09	420529.90	15750550.00
16	26710.23	81.54	9.50	2364.61	82.19	482354.90	16525380.00
18	31533.84	81.64	10.64	2459.00	82.27	541531.10	17208980.00
20	36535.78	81.73	11.79	2542.95	82.34	598148.40	17817190.00
22	41697.15	81.81	12.93	2618.42	82.41	652305.80	18365850.00
24	47002.18	81.88	14.07	2686.61	82.47	703944.60	18863880.00
26	52438.42	81.94	15.20	2749.63	82.54	754033.80	19326270.00
28	57997.13	82.00	16.33	2809.08	82.60	803449.20	19763530.00

Compartment 5 From 174 to 220 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	92.00	36.00	30.67	135454.70
2	1460.69	34.92	1.29	1368.69	34.87	102748.40	1941778.00
4	4556.08	34.97	2.48	1726.69	35.13	204838.70	2471081.00
6	8235.19	35.07	3.61	1952.43	35.28	295223.50	2807329.00
8	12310.56	35.15	4.74	2122.95	35.38	378818.40	3062801.00
10	16690.85	35.21	5.86	2257.34	35.46	454838.60	3265200.00
12	21313.26	35.27	6.98	2365.07	35.53	522595.70	3429095.00
14	26130.40	35.32	8.09	2452.07	35.60	581924.80	3563420.00
16	31106.14	35.37	9.19	2523.67	35.67	633955.40	3675896.00
18	36213.91	35.42	10.30	2584.10	35.73	680236.10	3771749.00
20	41435.54	35.46	11.39	2637.52	35.79	723033.60	3856776.00
22	46758.01	35.50	12.49	2684.95	35.83	762577.80	3931466.00
24	52172.25	35.53	13.58	2729.29	35.86	800878.70	4000560.00
26	57671.11	35.57	14.67	2769.57	35.89	836791.90	4062667.00
28	63248.60	35.60	15.76	2807.92	35.91	871983.20	4121711.00

Compartment 6 From 220 to 254 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	68.00	-4.00	22.67	7638.67
2	1176.53	-3.89	1.29	1108.53	-4.11	98272.91	124354.00
4	3673.93	-3.94	2.48	1388.87	-4.08	193167.00	156223.30
6	6622.00	-3.99	3.61	1559.20	-4.07	273312.40	175525.60
8	9862.79	-4.02	4.73	1681.60	-4.06	342863.20	189332.50
10	13318.11	-4.04	5.84	1773.72	-4.05	402343.30	199676.30
12	16935.49	-4.05	6.94	1843.66	-4.04	451821.30	207500.40
14	20674.84	-4.05	8.04	1895.69	-4.03	491140.00	213291.00
16	24505.81	-4.05	9.13	1935.29	-4.03	522546.00	217678.80
18	28407.82	-4.05	10.21	1966.72	-4.02	548412.60	221150.00
20	32368.73	-4.05	11.29	1994.20	-4.02	571708.90	224186.60
22	36382.91	-4.05	12.36	2019.98	-4.01	594162.30	227034.40
24	40448.08	-4.04	13.43	2045.19	-4.01	616689.40	229829.60
26	44563.65	-4.04	14.50	2070.38	-4.01	639754.80	232640.30
28	48728.85	-4.04	15.57	2094.82	-4.01	662674.60	235370.60

Compartment 7 From 254 to 300 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	92.00	-44.00	30.67	194334.70
2	1372.63	-42.24	1.29	1280.63	-42.98	86215.59	2588396.00
4	4387.36	-42.44	2.50	1734.11	-43.45	208768.80	3565696.00
6	8140.19	-42.73	3.66	2018.72	-43.68	326305.20	4190896.00
8	12379.42	-42.96	4.81	2220.51	-43.80	432538.60	4637423.00
10	16970.03	-43.13	5.95	2370.11	-43.87	525016.90	4970290.00
12	21819.74	-43.27	7.07	2479.60	-43.92	600732.40	5213524.00
14	26856.80	-43.37	8.18	2557.46	-43.94	658919.40	5385482.00
16	32028.21	-43.46	9.29	2613.95	-43.96	703474.30	5509234.00
18	37299.21	-43.52	10.38	2657.05	-43.97	738809.80	5602920.00
20	42649.99	-43.57	11.46	2693.73	-43.97	769811.40	5681889.00
22	48071.29	-43.61	12.53	2727.57	-43.97	799186.10	5754326.00
24	53559.17	-43.65	13.61	2760.31	-43.98	828305.90	5824093.00
26	59113.09	-43.68	14.68	2793.61	-43.98	858645.20	5894637.00
28	64733.34	-43.70	15.75	2826.64	-43.98	889451.40	5964863.00

Compartment 8 From 300 to 338 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	75.88	-85.99	25.22	570066.20
2	716.02	-83.05	1.26	640.14	-83.10	17879.27	4633161.00
4	2364.84	-83.17	2.53	1008.67	-83.89	65021.64	7371683.00
6	4677.95	-83.56	3.77	1304.45	-84.49	134601.30	9608064.00
8	7527.24	-83.92	5.00	1544.83	-84.93	218321.30	11443790.00
10	10817.18	-84.23	6.23	1745.11	-85.28	310500.10	12985270.00
12	14465.83	-84.50	7.43	1903.54	-85.54	400097.30	14214350.00
14	18384.83	-84.72	8.62	2015.45	-85.71	473451.00	15085640.00
16	22490.15	-84.90	9.79	2089.88	-85.81	527267.70	15663730.00
18	26721.97	-85.04	10.93	2141.94	-85.86	567414.30	16065820.00
20	31045.85	-85.16	12.05	2181.94	-85.88	599690.30	16372750.00
22	35442.89	-85.24	13.16	2215.11	-85.90	627393.40	16625830.00
24	39904.20	-85.32	14.26	2246.21	-85.91	654162.20	16861900.00
26	44425.98	-85.38	15.36	2275.56	-85.92	680125.00	17083940.00
28	49005.80	-85.42	16.45	2304.27	-85.92	706179.20	17300340.00

Compartment 9 From 338 to 370 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	56.16	-121.50	15.67	817582.10
2	271.94	-116.81	1.20	215.78	-120.40	1229.29	3067178.00
4	904.14	-116.64	2.53	416.41	-120.88	8089.92	5951544.00
6	1976.84	-117.04	3.91	656.29	-121.09	27980.62	9455176.00
8	3540.86	-117.64	5.30	907.74	-121.09	66921.55	13173830.00
10	5592.51	-118.22	6.67	1143.91	-121.06	127298.10	16681970.00
12	8091.10	-118.71	8.02	1354.68	-121.05	206554.70	19818720.00
14	10967.99	-119.10	9.33	1522.21	-121.03	289527.60	22322530.00
16	14127.47	-119.41	10.60	1637.27	-121.02	358497.90	24044860.00
18	17475.64	-119.65	11.83	1710.90	-121.02	408342.50	25145740.00
20	20948.06	-119.83	13.02	1761.51	-121.01	445365.00	25899630.00
22	24509.41	-119.97	14.18	1799.84	-121.01	474917.00	26469140.00
24	28141.09	-120.07	15.32	1831.84	-121.01	500615.30	26943490.00
26	31833.54	-120.16	16.44	1860.62	-121.01	524529.10	27369220.00
28	35580.99	-120.23	17.55	1886.83	-121.01	546969.90	27756440.00

Compartment 10 From 370 to 410 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	17.75	-139.51	2.24	394467.00
2	56.14	-139.49	1.12	38.39	-139.47	22.89	851311.60
4	175.58	-139.48	2.48	81.04	-139.48	214.74	1797554.00
6	444.84	-140.83	4.09	188.23	-143.02	1791.99	4261509.00
8	1039.88	-143.59	5.82	406.82	-147.82	9681.17	9457547.00
10	2161.02	-146.44	7.52	714.32	-151.39	32481.02	16924400.00
12	3953.26	-148.81	9.13	1077.92	-153.82	84256.59	25908970.00
14	6496.29	-150.78	10.66	1465.10	-155.35	176814.20	35744950.00
16	9723.88	-152.30	12.11	1762.50	-156.15	290985.50	43337310.00
18	13419.87	-153.34	13.46	1933.49	-156.45	379469.90	47684060.00
20	17382.00	-154.03	14.73	2028.64	-156.55	437066.90	50078960.00
22	21502.71	-154.49	15.93	2092.08	-156.59	478901.10	51665260.00
24	25735.73	-154.82	17.09	2140.95	-156.62	513005.60	52883340.00
26	30059.43	-155.06	18.23	2182.75	-156.63	543456.60	53924870.00
28	34463.60	-155.25	19.35	2221.42	-156.66	572650.90	54889340.00

Compartment 11 From 410 to 442 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	2.09	-177.00	5.33	2.09	-177.00	0.03	68876.20
8	14.83	-177.00	6.96	10.66	-177.00	4.57	351894.50
10	52.22	-177.00	8.52	26.74	-177.00	72.09	882867.60
12	181.41	-179.54	10.43	102.46	-188.42	684.01	3586603.00
14	731.79	-182.65	12.52	447.92	-195.17	17488.10	16205680.00
16	1998.12	-184.75	14.15	818.40	-195.02	74953.08	29956310.00
18	4067.50	-187.28	15.64	1250.99	-193.41	165991.50	46474050.00
20	6717.72	-189.09	16.97	1399.23	-193.21	226158.70	52083960.00
22	9588.22	-190.03	18.18	1471.28	-193.17	261761.90	54792650.00
24	12578.94	-190.57	19.33	1519.44	-193.16	287873.70	56597930.00
26	15657.50	-190.93	20.44	1559.11	-193.15	310744.30	58082840.00
28	18811.83	-191.17	21.54	1595.22	-193.14	332626.20	59433640.00

Compartment 12 From 442 to 466 feet aft FP

Draft (ft)	Volume (ft^3)	LCG (ft fwd amid)	KG (ft)	A_{WP} (ft^2)	LCF (ft fwd amid)	I_T (ft^4)	I_L (ft^4)
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.03	-209.00	11.33	0.03	-209.00	0.00	1318.07
14	0.27	-209.00	13.04	0.21	-209.00	0.01	9171.25
16	93.38	-209.67	15.33	92.91	-231.59	438.56	4527355.00
18	588.75	-209.68	16.91	402.46	-231.65	35617.38	19617750.00
20	1479.10	-209.68	18.19	487.89	-231.66	63452.25	23782090.00
22	2888.35	-212.90	19.61	921.36	-222.15	116890.50	45029610.00
24	4780.48	-215.66	20.95	970.77	-221.99	135604.50	47446940.00
26	6760.48	-216.93	22.14	1009.22	-221.88	151612.30	49327660.00
28	8812.61	-217.67	23.27	1042.92	-221.80	166747.20	50975570.00

Table 4: Hydrostatics Page

Table 5: CurveReader Page

Curve Interpolator

Draft for Hydrostatic Interpolation (ft) : 24.96

Upper Limit For Draft Interpolation (ft) = 26
Lower Limit For Draft Interpolation (ft) = 24

Hydrostatic Values for Upper and Lower Limits

Draft (ft)	Volume (ft^3)	Displacement (LTsw)	KB (ft)	BM_T (ft)	KM_T (ft)	LCB (ft fwd)	LCF (ft fwd)	BM_L (ft)	A_{WP} (ft^2)	TPI (LTsw/in)	MCT1" (LTsw)
24	359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
26	405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71

Linearly Interpolated Hydrostatic Values

Draft (ft)	Volume (ft^3)	Displacement (LTsw)	KB (ft)	BM_T (ft)	KM_T (ft)	LCB (ft fwd)	LCF (ft fwd)	BM_L (ft)	A_{WP} (ft^2)	TPI (LTsw/in)	MCT1" (LTsw)
24.96	381954.63	10912.99	15.14	14.97	30.10	-8.49	-21.91	822.27	22924.52	54.58	1584.83

COMPARTMENT 1

Upper Compt Volume Limit For CG Interpolation (ft^3) =	467.44	Water KG (ft) =	0.94
Lower Compt Volume Limit For CG Interpolation (ft^3) =	0.00	Water KG (ft) =	0.00
Interpolated KG of Flooded Water (ft above baseline) =	0.00	Water LCG (ft fwd) =	206.32
Interpolated LCG of Flooded Water (ft fwd amidships) =	0.00	Water LCG (ft fwd) =	0.00
Interpolated I_T of Flooded Water (ft^4) =	0.00	Water I_T (ft^4) =	681.99
		Water I_T (ft^4) =	2566.22

COMPARTMENT 2

Upper Compt Volume Limit For CG Interpolation (ft^3) =	257.51	Water KG (ft) =	1.20
Lower Compt Volume Limit For CG Interpolation (ft^3) =	0.00	Water KG (ft) =	0.00
Interpolated KG of Flooded Water (ft above baseline) =	0.00	Water LCG (ft fwd) =	172.83
Interpolated LCG of Flooded Water (ft fwd amidships) =	0.00	Water LCG (ft fwd) =	0.00
Interpolated I_T of Flooded Water (ft^4) =	0.00	Water I_T (ft^4) =	629.98
		Water I_T (ft^4) =	28.93

COMPARTMENT 3

Upper Compt Volume Limit For CG Interpolation (ft^3) =	591.48	Water KG (ft) =	1.25
Lower Compt Volume Limit For CG Interpolation (ft^3) =	0.00	Water KG (ft) =	0.00
Interpolated KG of Flooded Water (ft above baseline) =	0.00	Water LCG (ft fwd) =	127.47
Interpolated LCG of Flooded Water (ft fwd amidships) =	0.00	Water LCG (ft fwd) =	0.00
Interpolated I_T of Flooded Water (ft^4) =	0.00	Water I_T (ft^4) =	5677.06
		Water I_T (ft^4) =	19.00

COMPARTMENT 4

Upper Compt Volume Limit For CG Interpolation (ft^3) =	1064.18	Water KG (ft) =	1.27
Lower Compt Volume Limit For CG Interpolation (ft^3) =	0.00	Water KG (ft) =	0.00
Interpolated KG of Flooded Water (ft above baseline) =	0.00	Water LCG (ft fwd) =	80.33
Interpolated LCG of Flooded Water (ft fwd amidships) =	0.00	Water LCG (ft fwd) =	0.00
Interpolated I_T of Flooded Water (ft^4) =	0.00	Water I_T (ft^4) =	36250.52
		Water I_T (ft^4) =	31.92

Water I_T (ft^4) = 31.92

COMPARTMENT 5

Upper Compt Volume Limit For CG Interpolation (ft^3) = 1460.69
Lower Compt Volume Limit For CG Interpolation (ft^3) = 0.00
Interpolated KG of Flooded Water (ft above baseline) = 0.00
Interpolated LCG of Flooded Water (ft fwd amidships) = 0.00
Interpolated I_T of Flooded Water (ft^4) = 0.00

Water KG (ft) = 1.29
Water KG (ft) = 0.00
Water LCG (ft fwd) = 34.92
Water LCG (ft fwd) = 0.00
Water I_T (ft^4) = 102748.40
Water I_T (ft^4) = 30.67

COMPARTMENT 6

Upper Compt Volume Limit For CG Interpolation (ft^3) = 36382.91
Lower Compt Volume Limit For CG Interpolation (ft^3) = 32368.73
Interpolated KG of Flooded Water (ft above baseline) = 11.37
Interpolated LCG of Flooded Water (ft fwd amidships) = -4.05
Interpolated I_T of Flooded Water (ft^4) = 573556.41

Water KG (ft) = 12.36
Water KG (ft) = 11.29
Water LCG (ft fwd) = -4.05
Water LCG (ft fwd) = -4.05
Water I_T (ft^4) = 594162.30
Water I_T (ft^4) = 571708.90

COMPARTMENT 7

Upper Compt Volume Limit For CG Interpolation (ft^3) = 32028.21
Lower Compt Volume Limit For CG Interpolation (ft^3) = 26856.80
Interpolated KG of Flooded Water (ft above baseline) = 8.94
Interpolated LCG of Flooded Water (ft fwd amidships) = -43.43
Interpolated I_T of Flooded Water (ft^4) = 689636.13

Water KG (ft) = 9.29
Water KG (ft) = 8.18
Water LCG (ft fwd) = -43.46
Water LCG (ft fwd) = -43.37
Water I_T (ft^4) = 703474.30
Water I_T (ft^4) = 658919.40

COMPARTMENT 8

Upper Compt Volume Limit For CG Interpolation (ft^3) = 22490.15
Lower Compt Volume Limit For CG Interpolation (ft^3) = 18384.83
Interpolated KG of Flooded Water (ft above baseline) = 9.09
Interpolated LCG of Flooded Water (ft fwd amidships) = -84.80
Interpolated I_T of Flooded Water (ft^4) = 495116.69

Water KG (ft) = 9.79
Water KG (ft) = 8.62
Water LCG (ft fwd) = -84.90
Water LCG (ft fwd) = -84.72
Water I_T (ft^4) = 527267.70
Water I_T (ft^4) = 473451.00

COMPARTMENT 9

Upper Compt Volume Limit For CG Interpolation (ft^3) = 271.94
Lower Compt Volume Limit For CG Interpolation (ft^3) = 0.00
Interpolated KG of Flooded Water (ft above baseline) = 0.00
Interpolated LCG of Flooded Water (ft fwd amidships) = 0.00
Interpolated I_T of Flooded Water (ft^4) = 0.00

Water KG (ft) = 1.20
Water KG (ft) = 0.00
Water LCG (ft fwd) = -116.81
Water LCG (ft fwd) = 0.00
Water I_T (ft^4) = 1229.29
Water I_T (ft^4) = 15.67

COMPARTMENT 10

Upper Compt Volume Limit For CG Interpolation (ft^3) = 56.14
Lower Compt Volume Limit For CG Interpolation (ft^3) = 0.00
Interpolated KG of Flooded Water (ft above baseline) = 0.00
Interpolated LCG of Flooded Water (ft fwd amidships) = 0.00
Interpolated I_T of Flooded Water (ft^4) = 0.00

Water KG (ft) = 1.12
Water KG (ft) = 0.00
Water LCG (ft fwd) = -139.49
Water LCG (ft fwd) = 0.00
Water I_T (ft^4) = 22.89
Water I_T (ft^4) = 2.24

COMPARTMENT 11

Upper Compt Volume Limit For CG Interpolation (ft^3) = 2.09
Lower Compt Volume Limit For CG Interpolation (ft^3) = 0.00
Interpolated KG of Flooded Water (ft above baseline) = 0.00
Interpolated LCG of Flooded Water (ft fwd amidships) = 0.00
Interpolated I_T of Flooded Water (ft^4) = 0.00

Water KG (ft) = 5.33
Water KG (ft) = 0.00
Water LCG (ft fwd) = -177.00
Water LCG (ft fwd) = 0.00
Water I_T (ft^4) = 0.03
Water I_T (ft^4) = 0.00

COMPARTMENT 12

Upper Compt Volume Limit For CG Interpolation (ft^3) = 0.03
Lower Compt Volume Limit For CG Interpolation (ft^3) = 0.00
Interpolated KG of Flooded Water (ft above baseline) = 0.00
Interpolated LCG of Flooded Water (ft fwd amidships) = 0.00
Interpolated I_T of Flooded Water (ft^4) = 0.00

Water KG (ft) = 11.33
Water KG (ft) = 0.00
Water LCG (ft fwd) = -209.00
Water LCG (ft fwd) = 0.00
Water I_T (ft^4) = 0.00
Water I_T (ft^4) = 0.00

359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
359875.90	10282.17	14.57	15.42	29.98	-7.72	-22.39	854.12	22660.18	53.95	1554.32
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
405745.50	11592.73	15.75	14.48	30.23	-9.32	-21.39	787.95	23209.36	55.26	1617.71
Error	Error	Error	Error	Error	Error	Error	Error	Error	Error	Error

Compartment 1 Calculations

818.6701	467.4372	1.810743	0.9405366	205.4192	206.3237	378.788	681.9908
1139.169	818.6701	2.708633	1.810743	204.5408	205.4192	468.2545	378.788
1478.622	1139.169	3.697621	2.708633	203.7959	204.5408	705.2813	468.2545
1857.753	1478.622	4.783825	3.697621	203.1969	203.7959	1069.809	705.2813
2285.55	1857.753	5.951108	4.783825	202.7249	203.1969	1571.871	1069.809
2767.198	2285.55	7.181381	5.951108	202.3587	202.7249	2232.965	1571.871
3308.027	2767.198	8.46279	7.181381	202.0805	202.3587	3107.372	2232.965
3914.136	3308.027	9.787691	8.46279	201.8776	202.0805	4229.092	3107.372
4591.771	3914.136	11.14992	9.787691	201.7382	201.8776	5689.041	4229.092
5350.442	4591.771	12.54934	11.14992	201.6717	201.7382	7546.637	5689.041
6202.626	5350.442	13.98785	12.54934	201.6955	201.6717	9906.832	7546.637
7158.851	6202.626	15.46129	13.98785	201.8031	201.6955	12884.15	9906.832
8228.68	7158.851	16.96387	15.46129	201.9752	201.8031	16737.64	12884.15

Compartment 4 Calculations

3378.314	1064.182	2.493323	1.273245	80.41826	80.32667	93552.91	36250.52
6334.112	3378.314	3.676954	2.493323	80.68552	80.41826	156817.8	93552.91
9762.651	6334.112	4.851098	3.676954	80.92242	80.68552	223391.6	156817.8
13571.53	9762.651	6.019712	4.851098	81.12051	80.92242	290600.3	223391.6
17696.4	13571.53	7.18329	6.019712	81.28633	81.12051	356773.5	290600.3
22088.3	17696.4	8.341702	7.18329	81.42574	81.28633	420529.9	356773.5
26710.23	22088.3	9.495189	8.341702	81.54343	81.42574	482354.9	420529.9
31533.84	26710.23	10.64417	9.495189	81.64404	81.54343	541531.1	482354.9
36535.78	31533.84	11.78889	10.64417	81.73153	81.64404	598148.4	541531.1
41697.15	36535.78	12.92967	11.78889	81.8087	81.73153	652305.8	598148.4
47002.18	41697.15	14.06677	12.92967	81.87807	81.8087	703944.6	652305.8
52438.42	47002.18	15.20061	14.06677	81.94157	81.87807	754033.8	703944.6
57997.13	52438.42	16.33185	15.20061	82.00044	81.94157	803449.2	754033.8

Compartment 7 Calculations

32028.21	26856.8	9.285923	8.184948	-43.45575	-43.37316	703474.3	658919.4
32028.21	26856.8	9.285923	8.184948	-43.45575	-43.37316	703474.3	658919.4
32028.21	26856.8	9.285923	8.184948	-43.45575	-43.37316	703474.3	658919.4
32028.21	26856.8	9.285923	8.184948	-43.45575	-43.37316	703474.3	658919.4
32028.21	26856.8	9.285923	8.184948	-43.45575	-43.37316	703474.3	658919.4
32028.21	26856.8	9.285923	8.184948	-43.45575	-43.37316	703474.3	658919.4
32028.21	26856.8	9.285923	8.184948	-43.45575	-43.37316	703474.3	658919.4
37299.21	32028.21	10.37644	9.285923	-43.52056	-43.45575	738809.8	703474.3
42649.99	37299.21	11.45862	10.37644	-43.57211	-43.52056	769811.4	738809.8
48071.29	42649.99	12.53489	11.45862	-43.61378	-43.57211	799186.1	769811.4
53559.17	48071.29	13.60739	12.53489	-43.64804	-43.61378	828305.9	799186.1
59113.09	53559.17	14.67796	13.60739	-43.67657	-43.64804	858645.2	828305.9
64733.34	59113.09	15.74795	14.67796	-43.70072	-43.67657	889451.4	858645.2

Compartment 10 Calculations

175.5751	56.14449	2.4806	1.122526	-139.4807	-139.4861	214.7428	22.88841
444.841	175.5751	4.085932	2.4806	-140.8336	-139.4807	1791.993	214.7428
1039.883	444.841	5.823489	4.085932	-143.5898	-140.8336	9681.171	1791.993
2161.019	1039.883	7.518894	5.823489	-146.4384	-143.5898	32481.02	9681.171
3953.262	2161.019	9.12774	7.518894	-148.8138	-146.4384	84256.59	32481.02
6496.285	3953.262	10.66343	9.12774	-150.7824	-148.8138	176814.2	84256.59
9723.88	6496.285	12.11304	10.66343	-152.2982	-150.7824	290985.5	176814.2
13419.87	9723.88	13.46321	12.11304	-153.3393	-152.2982	379469.9	290985.5
17382	13419.87	14.72712	13.46321	-154.0262	-153.3393	437066.9	379469.9
21502.71	17382	15.93022	14.72712	-154.4906	-154.0262	478901.1	437066.9
25735.73	21502.71	17.09369	15.93022	-154.82	-154.4906	513005.6	478901.1
30059.43	25735.73	18.23138	17.09369	-155.0647	-154.82	543456.6	513005.6
34463.6	30059.43	19.35232	18.23138	-155.2539	-155.0647	572650.9	543456.6

Compartment 2 Calculations

774.26	257.51	2.45	1.20	170.9787	172.8288	2276.732	629.9792
1488.66	774.26	3.69	2.45	170.3857	170.9787	5043.934	2276.732
2381.36	1488.66	4.94	3.69	170.1049	170.3857	9054.294	5043.934
3443.04	2381.36	6.20	4.94	169.9538	170.1049	14473.58	9054.294
4666.80	3443.04	7.47	6.20	169.8729	169.9538	21232.65	14473.58
6045.40	4666.80	8.73	7.47	169.8345	169.8729	29342.89	21232.65
7571.92	6045.40	10.00	8.73	169.825	169.8345	38608.61	29342.89
9239.63	7571.92	11.27	10.00	169.8357	169.825	49062.61	38608.61
11043.75	9239.63	12.53	11.27	169.8603	169.8357	60788.6	49062.61
12981.69	11043.75	13.80	12.53	169.8955	169.8603	73940.39	60788.6
15052.63	12981.69	15.06	13.80	169.9387	169.8955	88802.46	73940.39
17256.62	15052.63	16.33	15.06	169.988	169.9387	105391.6	88802.46
19595.46	17256.62	17.61	16.33	170.0427	169.988	124355.1	105391.6

Compartment 5 Calculations

4556.077	1460.694	2.478391	1.291344	34.96865	34.91713	204838.7	102748.4
8235.192	4556.077	3.61407	2.478391	35.06651	34.96865	295223.5	204838.7
12310.56	8235.192	4.739589	3.61407	35.14711	35.06651	378818.4	295223.5
16690.85	12310.56	5.86036	4.739589	35.21392	35.14711	454838.6	378818.4
21313.26	16690.85	6.976727	5.86036	35.27132	35.21392	522595.7	454838.6
26130.4	21313.26	8.088227	6.976727	35.32297	35.27132	581924.8	522595.7
31106.14	26130.4	9.194603	8.088227	35.37112	35.32297	633955.4	581924.8
36213.91	31106.14	10.29607	9.194603	35.41656	35.37112	680236.1	633955.4
41435.54	36213.91	11.39335	10.29607	35.45928	35.41656	723033.6	680236.1
46758.01	41435.54	12.48722	11.39335	35.49876	35.45928	762577.8	723033.6
52172.25	46758.01	13.57848	12.48722	35.53459	35.49876	800878.7	762577.8
57671.11	52172.25	14.66774	13.57848	35.5668	35.53459	836791.9	800878.7
63248.6	57671.11	15.75544	14.66774	35.59572	35.5668	871983.2	836791.9

Compartment 8 Calculations

22490.15	18384.83	9.78721	8.621846	-84.90393	-84.72488	527267.7	473451
22490.15	18384.83	9.78721	8.621846	-84.90393	-84.72488	527267.7	473451
22490.15	18384.83	9.78721	8.621846	-84.90393	-84.72488	527267.7	473451
22490.15	18384.83	9.78721	8.621846	-84.90393	-84.72488	527267.7	473451
22490.15	18384.83	9.78721	8.621846	-84.90393	-84.72488	527267.7	473451
22490.15	18384.83	9.78721	8.621846	-84.90393	-84.72488	527267.7	473451
22490.15	18384.83	9.78721	8.621846	-84.90393	-84.72488	527267.7	473451
26721.97	22490.15	10.93011	9.78721	-85.04467	-84.90393	567414.3	527267.7
31045.85	26721.97	12.05447	10.93011	-85.15562	-85.04467	599690.3	567414.3
35442.89	31045.85	13.16456	12.05447	-85.24418	-85.15562	627393.4	599690.3
39904.2	35442.89	14.26443	13.16456	-85.31601	-85.24418	654162.2	627393.4
44425.98	39904.2	15.35734	14.26443	-85.37518	-85.31601	680125	654162.2
49005.8	44425.98	16.44559	15.35734	-85.42455	-85.37518	706179.2	680125

Compartment 11 Calculations

2.085751	0	5.333333	0	-177	0	0.03423032	0
2.085751	0	5.333333	0	-177	0	0.03423032	0
14.82779	2.085751	6.958227	2.085751	-177	-177	4.565001	0.03423032
52.2196	14.82779	8.522876	14.82779	-177	-177	72.09253	4.565001
181.4119	52.2196	10.42609	52.2196	-179.5424	-177	684.0118	72.09253
731.7929	181.4119	12.51929	181.4119	-182.6458	-179.5424	17488.1	684.0118
1998.116	731.7929	14.15326	731.7929	-184.7489	-182.6458	74953.08	17488.1
4067.502	1998.116	15.63702	1998.116	-187.2821	-184.7489	165991.5	74953.08
6717.718	4067.502	16.97111	4067.502	-189.0896	-187.2821	226158.7	165991.5
9588.222	6717.718	18.17978	6717.718	-190.0273	-189.0896	261761.9	226158.7
12578.94	9588.222	19.32709	9588.222	-190.5729	-190.0273	287873.7	261761.9
15657.5	12578.94	20.44334	12578.94	-190.9259	-190.5729	310744.3	287873.7
18811.83	15657.5	21.54339	15657.5	-191.173	-190.9259	332626.2	310744.3

Compartment 3 Calculations

1895.444	591.48	2.50	1.25	127.6306	127.4736	19919.67	5677.062
3698.192	1895.44	3.74	2.50	127.7686	127.6306	41817.94	19919.67
5923.647	3698.19	4.98	3.74	127.8948	127.7686	70313.69	41817.94
8521.65	5923.65	6.21	4.98	128.013	127.8948	104483.3	70313.69
11452.03	8521.65	7.44	6.21	128.1268	128.013	142563.3	104483.3
14678.37	11452.03	8.67	7.44	128.2367	128.1268	183194.1	142563.3
18167.42	14678.37	9.88	8.67	128.3417	128.2367	224773.3	183194.1
21891.65	18167.42	11.10	9.88	128.4411	128.3417	267508.3	224773.3
25831.09	21891.65	12.30	11.10	128.535	128.4411	311334.5	267508.3
29970.63	25831.09	13.51	12.30	128.6238	128.535	356571.4	311334.5
34298.58	29970.63	14.70	13.51	128.7076	128.6238	403153.4	356571.4
38806.83	34298.58	15.90	14.70	128.7872	128.7076	451997.5	403153.4
43490.43	38806.83	17.10	15.90	128.8631	128.7872	503373.4	451997.5

Compartment 6 Calculations

36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
36382.91	32368.73	12.35744	11.28538	-4.05	-4.05	594162.3	571708.9
40448.08	36382.91	13.42726	12.35744	-4.04	-4.05	616689.4	594162.3
44563.65	40448.08	14.49622	13.42726	-4.04	-4.04	639754.8	616689.4
48728.85	44563.65	15.56518	14.49622	-4.04	-4.04	662674.6	639754.8

Compartment 9 Calculations

904.1364	271.9445	2.531258	1.195658	-116.6428	-116.8092	8089.915	1229.289
1976.836	904.1364	3.911332	2.531258	-117.0437	-116.6428	27980.62	8089.915
3540.864	1976.836	5.299293	3.911332	-117.6393	-117.0437	66921.55	27980.62
5592.509	3540.864	6.670996	5.299293	-118.2239	-117.6393	127298.1	66921.55
8091.101	5592.509	8.016508	6.670996	-118.7112	-118.2239	206554.7	127298.1
10967.99	8091.101	9.328764	8.016508	-119.1022	-118.7112	289527.6	206554.7
14127.47	10967.99	10.5998	9.328764	-119.4108	-119.1022	358497.9	289527.6
17475.64	14127.47	11.82742	10.5998	-119.6476	-119.4108	408342.5	358497.9
20948.06	17475.64	13.01718	11.82742	-119.8276	-119.6476	445365	408342.5
24509.41	20948.06	14.17765	13.01718	-119.9657	-119.8276	474917	445365
28141.09	24509.41	15.31657	14.17765	-120.0741	-119.9657	500615.3	474917
31833.54	28141.09	16.44008	15.31657	-120.1609	-120.0741	524529.1	500615.3
35580.99	31833.54	17.55251	16.44008	-120.2318	-120.1609	546969.9	524529.1

Compartment 12 Calculations

0.03007398	0	11.33333	0	-209	0	1.85029E-05	0
0.03007398	0	11.33333	0	-209	0	1.85029E-05	0
0.03007398	0	11.33333	0	-209	0	1.85029E-05	0
0.03007398	0	11.33333	0	-209	0	1.85029E-05	0
0.03007398	0	11.33333	0	-209	0	1.85029E-05	0
0.2694061	0.03007398	13.03565	11.33333	-209	-209	0.006233247	1.85029E-05
93.38425	0.2694061	15.32521	13.03565	-209.6746	-209	438.5617	0.006233247
588.7519	93.38425	16.90962	15.32521	-209.6789	-209.6746	35617.38	438.5617
1479.103	588.7519	18.18718	16.90962	-209.6796	-209.6789	63452.25	35617.38
2888.347	1479.103	19.6096	18.18718	-212.9004	-209.6796	116890.5	63452.25
4780.477	2888.347	20.95498	19.6096	-215.6562	-212.9004	135604.5	116890.5
6760.475	4780.477	22.14157	20.95498	-216.9296	-215.6562	151612.3	135604.5
8812.613	6760.475	23.2742	22.14157	-217.6688	-216.9296	166747.2	151612.3

Table 5: CurveReader Page

Table 6: Trim Page

Trimming Calculation

Total Weight Added (LTsw) =	2375.96
LCG of added Weight (ft fwd) =	-37.91
VCG of added Weight (ft) =	-52.38
Location of fwd draft (ft fwd FP) =	2.97
Initial fwd draft (ft) =	21.50
Location of aft draft (ft aft AP) =	1.99
Initial aft draft (ft) =	21.50
TPI (LTsw/in) =	54.58
MCT1" (ft/LTsw) =	1584.83
LCF (ft fwd) =	-21.91
LBP (ft) =	466.00

Parallel Sinkage (ft) = 3.63
Parallel Sinkage = Weight Added / (TPI x 12 in/ft)

Trimming Moment (ft-LTsw) = -38029.16
Trimming Moment = Weight Added x (LCG of added Weight - LCF)

Trim between Perpendiculars (ft) = -2.00
Trim between Perpendiculars = Trimming Moment / (MCT1" x 12 in/ft)

Draft change at fwd mark due to trim (ft) = -1.11
Draft change at bow due to trim = (((LBP / 2) + Location of FP - LCF) / LBP) x Trim Bet Perpendiculars

Draft change at aft mark due to trim (ft) = -0.91
Draft change at stern due to trim = (((LBP / 2) + Location of AP + LCF) / LBP) x Trim Bet Perpendiculars

New Draft at bow (ft) = 24.02
New Draft at bow = Original Draft at bow + Parallel Sinkage + Draft change due to trim

New Draft at stern (ft) = 26.04
New Draft at stern = Original Draft at stern + Parallel Sinkage - Draft change due to trim

Table 6: Trim Page

Table 7: Holes Page

Hole Characteristics Compiler

SIMSMART DDE Poke out from blue boxes

Compartment	Long'l Hole Location (ft fwd amidships)	Vert Hole Location (ft above BL)	New Draft at Hole (ft)	New Hole Depth (ft)	Current Pressure at Hole (psi)	New Pressure at Ho (psi)
1	193.00	16.00	24.21	8.21	0.00	0.00
2	0.00	0.00	25.03	25.03	0.00	0.00
3	0.00	0.00	25.03	25.03	0.00	0.00
4	0.00	0.00	25.03	25.03	0.00	0.00
5	0.00	0.00	25.03	25.03	0.00	0.00
6	13.00	11.50	24.98	13.48	20.62	20.67
7	0.00	0.00	25.03	25.03	0.00	0.00
8	0.00	0.00	25.03	25.03	0.00	0.00
9	0.00	0.00	25.03	25.03	0.00	0.00
10	-157.00	18.00	25.71	7.71	0.00	0.00
11 ..	0.00	0.00	25.03	25.03	0.00	0.00
12	0.00	0.00	25.03	25.03	0.00	0.00

Table 7: Holes Page

Table 8: Perpendiculars Page

Location of Perpendiculars w.r.t Draft

Draft (ft)	Forward Draft Mark (ft fwd FP)	Aft Draft Mark (ft aft AP)
0	-1.00	-87.00
2	-2.00	-83.00
4	-3.00	-78.00
6	-3.00	-72.00
8	-3.50	-64.00
10	-3.00	-55.00
12	-2.75	-45.50
14	-2.25	-35.00
16	-1.50	-20.50
18	-1.00	-0.50
20	-0.50	0.00
22	1.50	1.00
24	3.00	1.50
26	4.50	2.00
28	6.00	2.50

FP Draft Mark Interpolator

Draft at FP (ft) =	23.97		
Upper Draft Limit =	24.00	Corresponding Mark Location =	3.00
Lower Draft Limit =	22.00	Corresponding Mark Location =	1.50
Location of Forward Draft Mark (ft fwd) =		2.97	

AP Draft Mark Interpolator

Draft at AP (ft) =	25.96		
Upper Draft Limit =	26.00	Corresponding Mark Location =	2.00
Lower Draft Limit =	24.00	Corresponding Mark Location =	1.50
Location of Aft Draft Mark (ft fwd) =		1.99	

Data Flow

24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
24	22	3	1.5	26	24	2	1.5
26	24	4.5	3	26	24	2	1.5
28	26	6	4.5	28	26	2.5	2

Table 8: Perpendiculars Page

Table 9: Free Surface Page

Free Surface Correction Monitor

Current BM_T (ft) =	14.97
Current KB (ft) =	15.14
Current KG (ft) =	23.36
Overall I_T of Floodwater (ft^4) =	597121.50
Current Underwater Volume of Ship (ft^3) (V_S) =	381954.63

$GM_{CORRECTED} = KB + BM_T - KG - (I_T / V_S)$

$GM_{CORRECTED}$ (ft) = $\boxed{5.17}$

Table 9: Free Surface Page

Table 10: KG Change Page

Change in KG due to Added Weight Calculator

$KG_{INITIAL}$ (ft) =	27.00
$D_{INITIAL}$ (LTsw) =	8696.15
Weighted KG of Flooded Water (ft-LTsw) =	23604.59
Weight of Flood Water (LTsw) =	2375.96

$KG_{NEW} = (D \;\; x\, KG_{CURRENT} \;\; x\, S(KG_{COMPARTMENT}\, x\, W_{FLOOD\ COMPARTMENT})) / (D + W)$

KG_{NEW} (ft) = $\boxed{23.34}$

Table 10: KG Change Page

Table 11: Curves Page

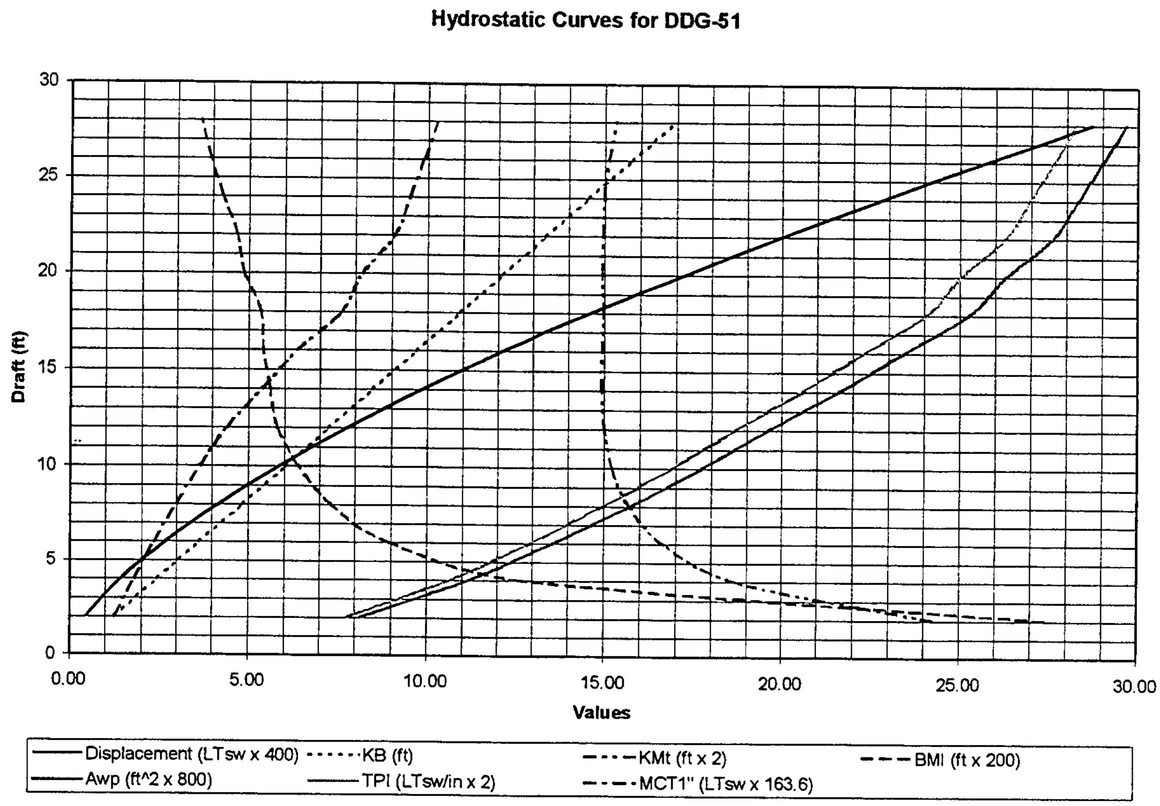

Table 11: Curves Page

Table 12: Flotation Page

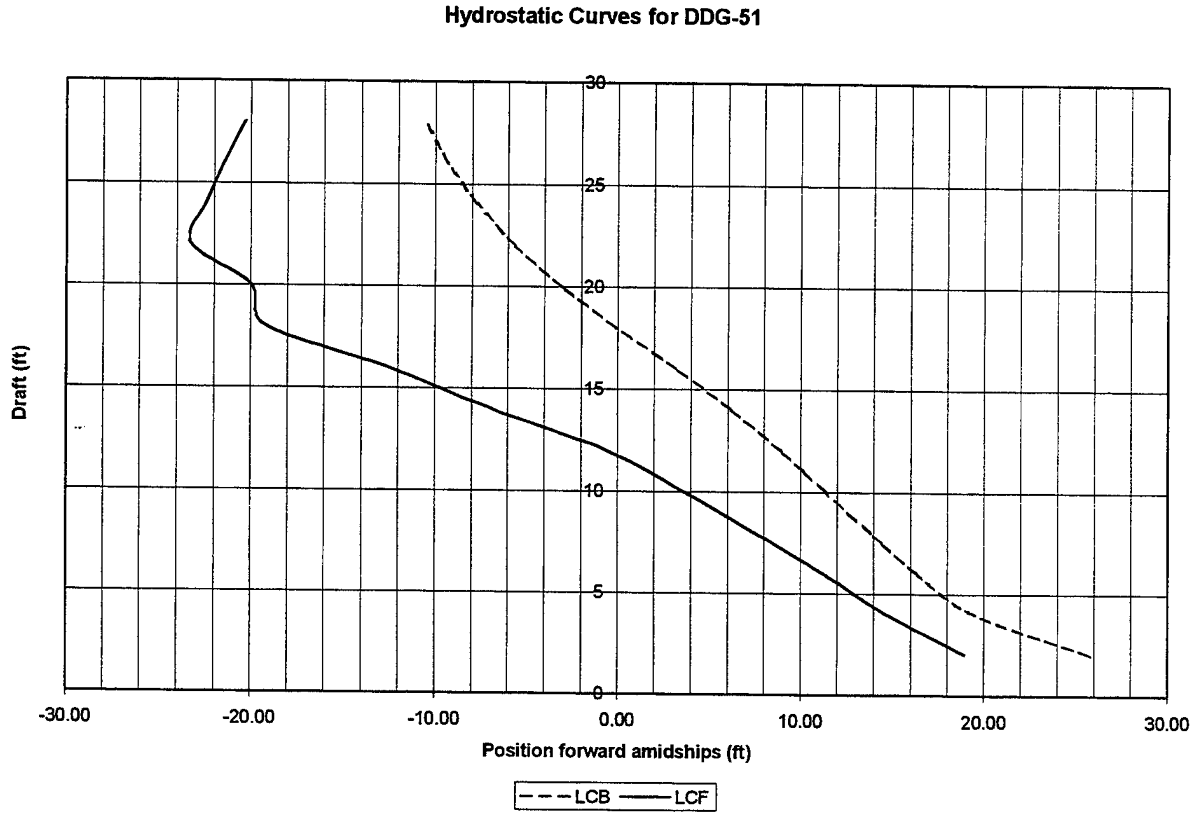

Table 12: Flotation Page

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F: EXCEL ERROR TABLES

Table 13: Waterplane area comparison between Excel and GHS

	Excel	GHS		
Waterline ---	A_{WP} (ft^2)	A_{WP} (ft^2)	Difference (ft^2)	Difference (%)
4	8975.10	8902	-73.10	0.82
6	10872.21	10819	-53.21	0.49
8	12594.64	12530	-64.64	0.52
10	14209.46	14141	-68.46	0.48
12	15739.40	15758	18.60	0.12
14	17251.35	17212	-39.35	0.23
16	18689.21	18827	137.79	0.73
18	20330.59	20079	-251.59	1.25
20	21199.05	21206	6.95	0.03
22	22110.72	22072	-38.72	0.18

Table 13

Table 14: Section area comparison between Excel and GHS

	Excel	GHS		
Station ---	Section Area (ft^2)	Section Area (ft^2)	Difference (ft^2)	Difference (%)
1	144.97	139.83	-5.14	3.68
2	271.03	260.27	-10.76	4.14
3	426.60	410.39	-16.21	3.95
4	574.58	553.93	-20.65	3.73
5	711.26	686.99	-24.27	3.53
6	830.60	803.83	-26.77	3.33
7	925.33	897.24	-28.09	3.13
8	996.30	967.42	-28.88	2.99
9	1046.58	1017.18	-29.40	2.89
10	1072.75	1043.01	-29.74	2.85
11	1073.27	1043.52	-29.75	2.85
12	1047.89	1018.27	-29.62	2.91
13	993.54	964.17	-29.37	3.05
14	909.03	880.09	-28.94	3.29
15	795.21	766.99	-28.22	3.68
16	655.35	628.28	-27.07	4.31
17	495.91	469.18	-26.73	5.70
18	336.38	309.98	-26.40	8.52
19	201.48	173.29	-28.19	16.27

Table 14

Tables 15-18: Hydrostatics comparison between DataInput.exe and GHS

	General HydroStatics Results							Data Input Program Results										
Draft (ft)	Displacement (LTsw)	LCB (ft aft FP)	KB (ft above BL)	A_{wl} (ft^2)	LCF (ft aft FP)	BM_T (ft)	BM_L (ft)	Volume (ft^3)	Displacement (LTsw)	LCB (ft fore CL)	LCB (ft aft FP)	KB (ft above BL)	A_{wl} (ft^2)	LCF (ft fore CL)	LCF (ft aft FP)	BM_T (ft)	BM_L (ft)
2	206.04	211.65	1.27	6452	215.65	47.94	5130.4	7434.56	212.29	25.77	207.23	1.25	6533.8	18.96	214.04	47.23	5477.67
4	645.52	215.69	2.49	8902	218.81	34.95	2443.3	22928.60	654.73	19.47	213.53	2.47	8960.2	14.67	218.38	34.87	2536.83
6	1211.30	218.11	3.68	10819	221.86	29.59	1734.7	42759.23	1220.99	16.36	216.64	3.66	10870.4	11.23	221.77	29.65	1788.64
8	1879.38	220.08	4.88	12530	225.40	26.47	1422.0	66221.85	1890.97	13.82	219.18	4.85	12592.3	7.41	225.59	26.54	1465.50
10	2642.52	222.18	6.08	14141	229.54	24.32	1253.2	92995.89	2655.51	11.38	221.62	6.05	14181.8	3.54	229.46	24.40	1278.66
12	3496.85	224.54	7.29	15758	235.01	22.73	1169.5	122860.90	3508.31	8.94	224.06	7.26	15683.2	-0.68	233.68	22.77	1162.40
14	4435.35	226.97	8.51	17212	239.79	21.32	1097.2	155822.50	4449.53	6.20	226.80	8.48	17278.4	-7.02	240.02	21.39	1121.75
16	5464.05	229.87	9.74	18827	247.03	20.21	1088.3	191830.70	5477.75	3.17	229.83	9.70	18729.9	-12.64	245.64	20.16	1081.04
18	6570.03	232.76	10.97	20079	251.18	18.97	1038.5	230773.90	6589.77	-0.07	233.07	10.94	20213.2	-19.21	252.21	19.03	1068.77
20	7749.73	235.64	12.20	21206	254.68	17.75	993.5	272029.40	7767.83	-3.03	236.03	12.16	21042.2	-19.97	252.97	17.71	974.60
22	8989.28	238.2	13.42	22072	256.08	16.50	931.5	315143.70	8998.96	-5.58	238.58	13.37	22072.1	-23.22	256.22	16.54	933.71
24	10267.08	240.44	14.61	22660	255.24	15.38	852.1	359875.90	10276.30	-7.72	240.72	14.57	22660.2	-22.39	255.39	15.42	854.12
26	11577.38	242.06	15.79	23209	254.25	14.45	786.1	405745.50	11586.11	-9.32	242.32	15.75	23209.4	-21.39	254.39	14.48	787.95
28	12918.60	243.27	16.96	23742	253.16	13.67	731.4	452697.30	12926.82	-10.51	243.51	16.92	23742.5	-20.30	253.30	13.70	733.12

Table 15

δDisplacement (LTsw)	% Difference %
-6.25	3.04
-9.21	1.43
-9.69	0.80
-11.59	0.62
-12.99	0.49
-11.46	0.33
-14.18	0.32
-13.70	0.25
-19.74	0.30
-18.10	0.23
-9.68	0.11
-9.22	0.09
-8.73	0.08
-8.22	0.06

δLCB (ft aft FP)	% Difference %
4.42	2.09
2.16	1.00
1.47	0.67
0.90	0.41
0.56	0.25
0.48	0.21
0.17	0.08
0.04	0.02
-0.31	0.13
-0.39	0.17
-0.30	0.13
-0.28	0.12
-0.26	0.11
-0.24	0.10

δKB (ft above BL)	% Difference %
0.02	1.37
0.02	0.86
0.02	0.61
0.03	0.61
0.03	0.48
0.03	0.44
0.03	0.40
0.04	0.38
0.03	0.30
0.04	0.32
0.05	0.37
0.04	0.29
0.04	0.27
0.04	0.26

Table 16

δA_{WL} (ft^2)	% Difference %
-81.8	1.27
-58.2	0.65
-51.4	0.47
-62.3	0.50
-40.8	0.29
74.8	0.47
-66.3	0.39
97.1	0.52
-134.2	0.67
163.8	0.77
-0.1	0.00
-0.2	0.00
-0.4	0.00
-0.5	0.00

δLCF (ft aft FP)	% Difference %
1.61	0.75
0.43	0.20
0.09	0.04
-0.19	0.09
0.08	0.03
1.33	0.57
-0.23	0.10
1.39	0.56
-1.03	0.41
1.71	0.67
-0.14	0.05
-0.15	0.06
-0.14	0.05
-0.14	0.06

δBM_T (ft)	% Difference %
0.71	1.47
0.08	0.24
-0.06	0.20
-0.07	0.27
-0.08	0.32
-0.04	0.19
-0.07	0.34
0.05	0.23
-0.06	0.32
0.04	0.24
-0.04	0.21
-0.04	0.24
-0.03	0.21
-0.03	0.23

Table 17

δBM_L (ft)	% Difference %
-347.27	6.77
-93.53	3.83
-53.94	3.11
-43.50	3.06
-25.46	2.03
7.10	0.61
-24.55	2.24
7.26	0.67
-30.27	2.91
18.90	1.90
-2.21	0.24
-2.02	0.24
-1.85	0.24
-1.72	0.24

Table 18

Tables 19-21: Hydrostatic comparison between NPSHS and GHS

	NPSHS				GHS			
Draft (ft)	Volume (ft^3)	Displacement (LTsw)	LCG (ft fwd amid)	KG (ft)	Displacement (LTsw)	LCG (ft aft FP)	LCG (ft fwd amid)	KG (ft)
0.00	0.00	0.00		0.00	0.00			0.00
2.00	790.13	22.58	177.36	1.12				
4.00	2052.58	58.65	170.07	2.31				
4.38					64.99	63.91	169.09	2.59
6.00	3702.77	105.79	166.67	3.52				
6.97					129.85	66.67	166.33	4.15
8.00	5725.56	163.59	164.85	4.76				
10.00	8107.60	231.65	163.80	6.01				
10.87					259.65	68.42	164.58	6.56
12.00	10834.55	309.56	163.18	7.27				
13.99					389.57	68.98	164.02	8.51
14.00	13890.05	396.86	162.80	8.54				
16.00	17258.38	493.10	162.59	9.80				
16.68					518.96	69.18	163.82	10.20
18.00	20925.96	597.88	162.49	11.07				
19.12					649.29	69.23	163.77	11.74
20.00	24884.19	710.98	162.47	12.33				
21.35					778.98	69.20	163.80	13.15
22.00	29128.27	832.24	162.51	13.59				
23.43					908.68	69.13	163.87	14.47
24.00	33656.06	961.60	162.60	14.86				
25.38					1038.63	69.03	163.97	15.72
26.00	38468.06	1099.09	162.72	16.13				
27.23					1168.53	68.91	164.09	16.90
28.00	43568.80	1244.82	162.87	17.40				
28.13					1233.77	68.85	164.15	17.47
28.81					1272.53	68.81	164.19	17.81

Table 19: Forward Compartment

	NPSHS				GHS			
Draft (ft)	Volume (ft^3)	Displacement (LTsw)	LCG (ft fwd amid)	KG (ft)	Displacement (LTsw)	LCG (ft aft FP)	LCG (ft fwd amid)	KG (ft)
0.00	0.00	0.00	0.00	0.00				
2.00	2051.06	58.60	2.88	1.29				
3.16					125.58	230.14	2.86	2.00
4.00	6402.16	182.92	2.80	2.48				
5.00					250.42	230.20	2.80	3.05
6.00	11541.05	329.74	2.73	3.61				
8.00	17195.66	491.30	2.70	4.73				
8.21					501.63	230.26	2.74	4.85
10.00	23231.99	663.77	2.68	5.84				
11.12					752.18	230.26	2.74	6.46
12.00	29559.52	844.56	2.68	6.95				
13.86					1002.98	230.26	2.74	7.97
14.00	36108.93	1031.68	2.69	8.05				
16.00	42827.52	1223.64	2.70	9.14				
16.51					1253.99	230.24	2.76	9.41
18.00	49679.02	1419.40	2.72	10.22				
19.10					1504.99	230.22	2.78	10.81
20.00	56641.66	1618.33	2.74	11.30				
21.63					1755.71	230.20	2.80	12.18
22.00	63704.30	1820.12	2.75	12.38				
24.00	70861.77	2024.62	2.77	13.45				
24.13					2006.54	230.19	2.81	13.52
26.00	78111.58	2231.76	2.79	14.52				
26.58					2257.36	230.17	2.83	14.83
27.79					2382.80	230.16	2.84	15.48
28.00	85450.99	2441.46	2.80	15.59				
28.61					2458.01	230.16	2.84	15.87

Table 20: Amidships Compartment

	NPSHS				GHS			
Draft (ft)	Volume (ft^3)	Displacement (LTsw)	LCG (ft fwd amid)	KG (ft)	Displacement (LTsw)	LCG (ft aft FP)	LCG (ft fwd amid)	KG (ft)
0.00	0.00	0.00	0.00	0.00				
2.00	76.06	2.17	-138.14	1.13				
4.00	239.57	6.84	-138.09	2.49				
6.00	595.30	17.01	-139.23	4.06				
8.00	1345.71	38.45	-141.61	5.77				
10.00	2713.59	77.53	-144.18	7.44				
10.62					89.00	380.45	-147.45	8.26
12.00	4908.43	140.24	-146.87	9.06				
13.02					177.46	383.76	-150.75	10.19
14.00	8331.71	238.05	-150.64	10.71				
15.86					355.48	389.17	-156.17	12.45
16.00	13227.90	377.94	-154.71	12.32				
17.90					533.79	392.90	-159.90	13.97
18.00	19547.68	558.51	-158.64	13.84				
19.64					711.51	395.49	-162.49	15.18
20.00	26853.62	767.25	-161.71	15.25				
21.29					889.47	397.17	-164.17	16.24
22.00	34595.88	988.45	-163.71	16.54				
22.89					1067.54	398.33	-165.33	17.22
24.00	42606.04	1217.32	-165.07	17.75				
24.46					1245.50	399.18	-166.18	18.15
25.99					1423.50	399.82	-166.82	19.03
26.00	50815.42	1451.87	-166.03	18.92				
27.50					1601.38	400.33	-167.33	19.89
28.00	59198.72	1691.39	-166.76	20.07				
28.27					1690.25	400.54	-167.54	20.31
28.90					1743.57	400.66	-167.66	20.57

Table 21: Aft Compartment

LIST OF REFERENCES

1. Sarchin, T. H. and L. L. Goldberg, "Stability and Buoyancy Criteria for U.S. Naval Surface Ships," *SNAME Transactions,* Vol. 70, 1962.

2. Surko, S. W., "An Assessment of Current Warship Damaged Stability Criteria," *Naval Engineers Journal,* May 1994.

3. Lewis, E. V., *Principles of Naval Architecture, Volume I, 2nd Revision,* The Society of Naval Architects and Marine Engineers, Jersey City, New Jersey, 1988.

4. Naval Ship Engineering Center, *Design Data Sheet (DDS) 079-1, Stability and Buoyancy of U.S. Naval Surface Ships,* Hyattsville, Maryland, June 1976.

5. Zahn, P. B., Rosborough, J. and R. Carlstrom, "Shipboard Damaged Stability Assessment: The Flooding Casualty Control Software," *Naval Engineers Journal,* May 1993.

6. Anderson, T. J., *Development of a SIMSMART Based, Progressive Flooding Design Tool,* Thesis, Naval Postgraduate School, March 1999.

7. Rawson, K. J. and E. C. Tupper, *Basic Ship Theory, Volume 1, 4th Edition,* Longman Scientific and Technical, Essex, England, 1994.

8. Lindeburg, M. R., *Mechanical Engineering Reference Manual for the PE Exam, 10th Edition,* Professional Publications Inc., Belmont, California, 1997.

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

Anderson, T. J., *Development of a SIMSMART Based, Progressive Flooding Design Tool,* Thesis, Naval Postgraduate School, March 1999.

Bradley, J. C. and A. C. Millspaugh, *Programming in Visual Basic 6.0,* McGraw Hill, Boston, Massachusetts, 1999.

Ferreiro, L. D. and M. H. Stonehouse, "A Comparative Study of US and UK Frigate Design," *SNAME Transactions,* Vol. 99, 1991.

Jones, H. D., *DD-963 Intact and Damage Stability in Beam Waves and Wind,* David Taylor Research Center, Bethesda, Maryland, September 1991.

Jones, H. D., *DDG-51 Intact and Damage Stability in Beam Waves and Wind,* David Taylor Research Center, Bethesda, Maryland, January 1992.

Lewis, E. V., *Principles of Naval Architecture, Volume I, 2nd Revision,* The Society of Naval Architects and Marine Engineers, Jersey City, New Jersey, 1988.

Lindeburg, M. R., *Mechanical Engineering Reference Manual for the PE Exam, 10th Edition,* Professional Publications Inc., Belmont, California, 1997.

Naval Ship Engineering Center, *Design Data Sheet (DDS) 079-1, Stability and Buoyancy of U.S. Naval Surface Ships,* Hyattsville, Maryland, June 1976.

Perry, Greg, *Teach Yourself Visual Basic 6 in 21 Days, Professional Reference Edition,* Sams Publishing, Indianapolis, Indiana, 1999.

Rawson, K. J. and E. C. Tupper, *Basic Ship Theory, Volume 1, 4th Edition,* Longman Scientific and Technical, Essex, England, 1994.

Robertson Jr., J. B., G. C. Nickum, R. I. Price and E. M. Middleton, "The New Equivalent International Regulations on Subdivisions and Stability of Passenger Ships," *SNAME Transactions,* Vol. 82, 1974.

Sarchin, T. H. and L. L. Goldberg, "Stability and Buoyancy Criteria for U.S. Naval Surface Ships," *SNAME Transactions,* Vol. 70, 1962.

Surko, S. W., "An Assessment of Current Warship Damaged Stability Criteria," *Naval Engineers Journal,* May 1994.

Zahn, P. B., Rosborough, J. and R. Carlstrom, "Shipboard Damaged Stability Assessment: The Flooding Casualty Control Software," *Naval Engineers Journal,* May 1993.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   FT. Belvoir, VA, 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA, 93943-5101

3. Professor Charles Calvano, Code ME/CA.. 2
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA, 93943-5101

4. Professor Fotis Papoulias, Code ME/PA.. 1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA, 93943-5101

5. Naval/Mechanical Engineering Curricular Office (Code 34).. 1
   Naval Postgraduate School
   Monterey, CA, 93943-5101

6. Dr. Gregory White.. 1
   Department of Naval Architecture, Ocean and Marine Engineering
   US Naval Academy, Mail Stop 11d
   590 Holloway Road
   Annapolis, MD 21402-5042

7. ENS Keith S. Kulow 1
   15 Leonardo Dr.
   Clifton Park, NY 12065

8. Mr. Bob Keane.. 1
   c/o Commander Carderock Division NSWC
   500 MacArthur Blvd
   West Bethesda, MD 20817-5700

9. Mr. Al Brauer.. 1
   c/o Naval Sea Systems Command
   Code 03D
   2531 Jefferson Davis Highway
   Arlington, VA 22242-5160

10. Mr. Robert R. Wunderlick 1
c/o Commander Carderock Division NSWC
500 MacArthur Blvd
West Bethesda, MD 20817-5700

11. Mr. Michel Masse 1
President, AHT (Applied High Technology)
P.O. Box 385, Station B
Montreal, (QC) Canada H3B3J7